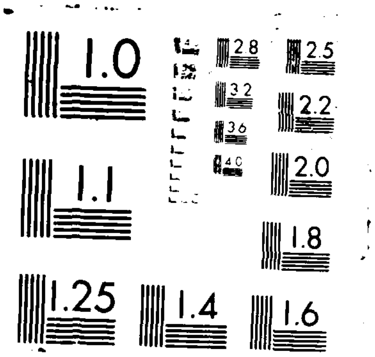
AD-A190 577 AN APPROXIMATION TECHNIQUE FOR SOLVING A SYSTEM OF 1/2
FREDHOLM INTEGRAL EQUA..(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. R B AANIEL
UNCLASSIFIED MAR 88 AFIT/GNE/ENG/88M-3 F/G 12/5 NL


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

DTIC FILE COPY

1

AD-A190 577

AN APPROXIMATION TECHNIQUE FOR SOLVING A SYSTEM OF FREDHOLM INTEGRAL EQUATIONS FOR ASYMMETRIC DETECTOR RESPONSE FUNCTIONS

THESIS

Russell B. Daniel
Captain, USMC

AFIT/GNE/ENG/88M-3

DTIC
ELECTE
APR 0 1988
S D
E

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

This document has been approved for public release and sale; its distribution is unlimited.

88 3 30 060

AFIT/GNE/ENG/88M-3

AN APPROXIMATION TECHNIQUE FOR SOLVING A SYSTEM OF FREDHOLM INTEGRAL EQUATIONS FOR ASYMMETRIC DETECTOR RESPONSE FUNCTIONS

THESIS

Russell B. Daniel
Captain, USMC

AFIT/GNE/ENG/88M-3

01

8

Approved for public release; distribution unlimited

AFIT/GNE/ENG/88M-3

AN APPROXIMATION TECHNIQUE FOR SOLVING A SYSTEM OF FREDHOLM INTEGRAL EQUATIONS FOR ASYMMETRIC DETECTOR RESPONSE FUNCTIONS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Nuclear Engineering (Effects)

Russell B. Daniel, B.S.

Captain, USMC

March 1988

Accession For
NTIS GRA&I ☒
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution/
Availability Codes
Avail and/or
Dist Special
A-1

Approved for public release, distribution unlimited

Preface

This study was designed to research and develop the methodology necessary to implement a general computer program that determines an approximate solution to a system of Fredholm integral equations. The computer program was designed to be easily converted into an excellent research and analysis tool for the Defense Nuclear Agency's underground nuclear effects simulation testing program.

Due to the symbols used during the study, two lists of symbols are provided. The first List of Symbols contains the symbols used to develop the methodology for this study. The second list is located in Tab 1 of Appendix C and contains all the variables used in the development of the computer program.

In conducting this study, I received guidance and strength from a number of people. First, I would like to thank my thesis advisor LCDR. Kirk A. Mathews for the assistance and knowledge he provided during my study. I would also like to thank my thesis committee; Dr. George John, Maj. Jim A. Lupo, and Lt. Col. Albert Alexander (from the Defense Nuclear Agency) for their assistance in compiling and completing this final copy of my thesis. Also a special thanks to LCDR. Mathews and Lt. Col. Alexander for their assistance in arranging my assignment to Field Command Defense Nuclear Agency as radiation diagnostician based on the knowledge gained during this study. Lastly, I would like to thank my wife, Karen, for her needed assistance in typing as well as moral support during this study.

Russell B. Daniel

Table of Contents

Page

Preface ii

List of Figures v

List of Tables. ix

List of Symbols x

Abstract xi

I. Introduction 1-1

Background. 1-1
Problem 1-2
Scope 1-3
Assumptions 1-3
General Approach. 1-4
Sequence of Presentation. 1-4

II. Detailed Problem Analysis 2-1

Introduction. 2-1
Errors Introduced During Unfolding. 2-1
Errors in Actual Practice versus
Simulated Errors in This Study. 2-2
Current Unfolding Techniques. 2-3

III. Theoretical Development 3-1

Introduction. 3-1
Response Functions. 3-1
Definition of Spectra. 3-5
Flux Non-Negativity Constraint. 3-7
Definition of Signals 3-7
Formulation of χ^2 Test Statistic. 3-8
Minimization of χ^2 Test Statistic 3-9
Normal Distribution for Simulating
Measurement Error. 3-12

IV. Development of the Computer Program 4-1

Introduction. 4-1
Development of Test Spectra 4-1
Planckian Unfolding Code. 4-1
Simulated Measurement Error 4-3
Cubic Spline Basis Function 4-3
Decon7.f 4-4

V. Results and Discussion. 5-1

Introduction 5-1
Validation Cases 5-1
Test Cases 5-5
Figure of Merit. 5-16

VI. Summary and Conclusions 6-1

Summary 6-1
Conclusions. 6-2

VII. Recommendations 7-1

Recommendations 7-1

Appendix A: Derivation of the Response Functions A-1

Appendix B: Derivation of the Cubic Spline Basis
Functions. B-1

Appendix C: Computer Program Documentation C-1

Appendix D: Computer Program Source Code D-1

Appendix E: Validation Results E-1

Appendix F: Test Case Results. F-1

Appendix G: Figure of Merit Study G-1

Bibliography. Bib.1

Vita. V.1

List of Figures

Figure Page

1. Linear Plot of the Closed Response Function 3-2

2. Linear Plot of the Open Response Function 3-4

3. Linear Plot of the (Planckian) Actual and (Planckian) Unfolded Spectra Versus Energy for the NP Cases. 5-7

4. Semi-log Plot of the (Planckian) Actual and (Planckian) Unfolded Spectra Versus Energy for the NP Cases. 5-8

5. Linear Plot of the (Spline) Actual and (Spline) Unfolded Spectra Versus Energy for the NC Cases. 5-9

6. Semi-log Plot of the (Spline) Actual and (Spline) Unfolded Spectra Versus Energy for the NC Cases. 5-10

7. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case PC3 5-13

8. Semi-log Plot of the Actual and Unfolded Spectra Versus Energy for Case PC3 5-14

9. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case CP3 5-17

10. Semi-log Plot of the Actual and Unfolded Spectra Versus Energy for Case CP3 5-18

11. Development of the Cubic Spline Basis Function B-3

12. Cubic Basis Functions for Case Using 6 Knots B-5

13. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case BC1 . . . . E-3

14. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case BC2 . . . . E-4

15. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case BC3 . . . . E-5

16. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case BC4 . . . . E-6

17. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case BF5 . . . . E-7

18. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case BF6 . . . . E-8

19. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case BF7 . . . . E-9

20. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case BF8 . . . . E-10

21. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP0 . . . . F-2

22. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP01. . . . . F-3

23. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP02. . . . . F-4

24. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP03. . . . . F-5

25. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP04. . . . . F-6

26. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP05. . . . . F-7

27. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP06. . . . . F-8

28. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP07. . . . . F-9

29. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP08. . . . . F-10

30. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP09. . . . . F-11

31. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP10 . . . . . F-12

32. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC0 . . . . . F-13

33. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC01 . . . . . F-14

34. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC02 . . . . . F-15

35. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC03 . . . . . F-16

36. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC04 . . . . . F-17

37. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC05 . . . . . F-18

38. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC06 . . . . . F-19

39. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC07 . . . . . F-20

40. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC08 . . . . . F-21

41. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC09 . . . . . F-22

42. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC10 . . . . . F-23

43. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case PC1 . . . . . F-24

44. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case PC2 . . . . . F-25

45. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case PC3 . . . . . F-26

46. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case CP1 . . . . F-27

47. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case CP2 . . . . F-28

48. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case CP3 . . . . F-29

49. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case CP4 . . . . F-30

List of Tables

Table Page

I. K-Edges for the 20 Detectors. 3-5

II. Comparison of the Unfolded Signals Using Varying Energy Bin Widths 3-10

III. Validation Cases for Planckian Basis Functions 5-2

IV. Validation Cases for Cubic Spline Basis Functions with Fixed Knots 5-3

V. Validation Cases for Cubic Spline Basis Functions with Variable Knots 5-4

VI. Noise Study for Planckian Basis Functions 5-6

VII. Noise Study for Cubic Spline Basis Functions with Variable Knots 5-11

VIII. Results of Fitting a Planckian Spectrum with Cubic Spline Basis Functions with Variable Knots. 5-15

IX. Results of Fitting a Cubic Spline Spectrum with Planckian Basis Functions. 5-19

X. Knots and Functions Used to Define the Cubic Spline Basis Functions. B-6

XI. List of Input Variables. C-1-1

XII. Continuation of the One Planckian Basis Function Validation Test E-1

XIII. Continuation of the Validation Cases for the Cubic Spline Basis Functions with Fixed Knots E-2

XIV. Comparison of Figures of Merit. G-3

List of Symbols

Symbol	Definition
a_j	coefficient for jth basis function
b_i	measured-to-predicted ratio
c_i	unfolded-to-predicted ratio
E	energy
E_i^0	k-edge of fluorescer for the ith detector
E_i^1	k-edge of filter for the ith detector
$F_j(E,P)$	jth basis function
ni	number of detectors
nj	number of basis functions
$R^c(E)$	closed response function
$R_i(E)$	calibrated response function
$R^o(E)$	open response function
$S_a(E)$	actual spectrum
$S_p(E)$	predicted spectrum
$S_u(E)$	unfolded spectrum
$S_u'(E)$	unfolded spectrum with flux non-negativity constraint
U_j	jth pseudo-random number
σ_b	standard deviation of b_i
χ^2	chi squared
Y_i^a	actual signal of the ith detector
Y_i^m	measured signal of the ith detector
Y_i^p	predicted signal of the ith detector
Y_i^u	unfolded signal of the ith detector
Z_i	simulated measurement added to b_i

AFIT/GNE/ENG/88M-3

Abstract

The purpose of this study was to develop the methodology for and to implement a computer program to approximate a solution to a system of Fredholm integral equations. The system of equations used in this study is representative of the equations formed during the detection of pulsed radiation using a series of detectors with asymmetric response functions. Though general in nature and applicable to all systems of Fredholm integral equations, the equations studied are of importance to the Defense Nuclear Agency with regard to the measurement of radiation spectra during underground nuclear effects simulation testing.

The deconvolution or solution technique consisted of representing the unfolded spectrum as a weighted sum of basis functions. This unfolded spectrum, the actual spectrum, and a predicted spectrum were then used to form a χ^2 test statistic. By adjusting the parameters in the basis functions and their weights, χ^2 was minimized and the unfolded spectrum was corrected to approximate the actual spectrum.

The methodology for this deconvolution technique was then converted into a general computer program. The validation cases conducted on the two types of spectra confirmed the reliability of the methodology and the computer program. Additionally, an initial study with simulated measurement error added to the measured-to-predicted ratios showed that the actual spectrum could not be returned exactly. The second study approximated the actual spectrum with an unfolded spectrum using a second set of basis functions. An acceptable approximation was conducted; however, certain artifacts were discovered in the unfolded spectrum.

The validation cases and preliminary test cases conducted prove that the computer program based on the methodology presented in this study is a viable means of approximating an actual radiation spectrum. Using this study and computer program as a starting point, the study of new basis functions and the effect of how well the actual spectrum can be approximated based on the number of detectors available to determine the spectrum is recommended.

AN APPROXIMATION TECHNIQUE FOR SOLVING A SYSTEM OF FREDHOLM INTEGRAL EQUATIONS FOR ASYMMETRIC DETECTOR RESPONSE FUNCTIONS

I. Introduction

Background

Approximate solutions to systems of Fredholm integral equations are needed by the Air Force in many fields of study, ranging from acoustics to optics. Such solutions also occur in disciplines as varied as geology and astronomy. An instance of particular importance to the Air Force arises in the measurement of radiation spectra emitted by nuclear devices detonated underground by the Defense Nuclear Agency for simulation of nuclear weapons effects. This study develops an approximate solution for a set of Fredholm integral equations of type 1 of the following form:

$$Y_i^m = \int_0^\infty S_a(E) R_i(E) dE \tag{1}$$

where

Y_i^m = the measured signal of the ith detector
$S_a(E)$ = the actual spectrum as a function of energy
$R_i(E)$ = the calibrated response function of the ith detector as a function of energy

In the detection of pulsed radiation, a set of detectors covering various energy ranges is used to obtain a set of measured signals. Each of these measured signals can be represented by an equation of the form of Equation (1). Since the response functions of these detectors are not rectangular in shape with neglible width and since a finite number of detectors must be used, only a limited resolution of the actual spectrum can be achieved. In order to achieve a reasonable resolution a filter-fluorescer detection system is utilized. The detectors and their response functions are discussed in detail in Section II.

Also, the experimenter does not know the actual spectrum or the exact response functions of the detectors. However, the experimenter can calibrate the detectors used and produce calibrated response functions for the detectors. The experimenter can also predict the shape of the actual spectrum based on source design and previous measurements. Thus, at the end of the experiment, the experimenter has the measured signals, a predicted spectrum, and the calibrated response functions of the detectors to use in approximating the actual spectrum. The procedure used to conduct this approximation is called deconvolution or unfolding.

Problem

The measured signals discussed above are the best information the experimenter has concerning the ideal signal (the error free signals from the actual spectrum.) These measured signals contain recording error and transmission error in addition to the error due to the response functions. These errors are discussed in detail in Section II. The main problem addressed in this study was the development of the methodology necessary to conduct a general deconvolution of the actual spectrum from the set of calibrated response functions and the set of measured signals noted in Equation (1). In other words, the known measured signals, calibrated response functions, and the predicted spectrum are used to determine an unfolded spectrum. When this unfolded spectrum is folded with the calibrated response functions, an approximation to the measured signals are returned. This unfolded spectrum is then used as an approximation to the actual spectrum. Secondly, a computer program was developed to conduct this deconvolution using various basis functions to construct the unfolded spectrum.

Scope

Since unclassified experimental data was unavailable, this study considered spectra that were constructed of either normalized planckian black body distributions, called planckians, or of cubic splines. During the study, a trial spectrum was constructed and used as the actual spectrum. This spectrum was then used to calculate the ideal signals or measured signals if simulated measurement errors are introduced. First, a study was conducted to determine if the basis functions could be used to unfold an actual spectrum produced from the same type of basis functions with no error introduced. Then a study was conducted to demonstrate the effects of simulated measurement error in the measured-to-predicted ratios on the unfolded spectrum. This simulated error represented the errors between the ideal signal and the measured signal. Finally, a study was conducted to determine if the basis functions could be used to unfold an actual spectrum constructed from another set of basis functions. The energy range considered in this study was from E_1^0 to $128E_1^0$, where E_1^0 (the k edge of the fluorescer for the first detector) is used as a convenient arbitrary energy unit.

Assumptions

In the development of the methodology, a basic assumption was that the nonuniqueness of the actual spectrum and the errors developed in the mathematics of an analytic solution required the approximation technique to be numerical. Also in order to define or bound the study and to allow a test method to be developed, the following assumptions are made:

1. Twenty detectors are available to conduct the experiment. Thirteen detectors have a closed response function and seven detectors have an open response function, as described in Section III.
2. A resolving power (E center/ Δ E) of about 1.5 is desired.

3. The exact response functions are equal to the calibrated response functions, as described in Section II. (This does not restrict the applicability of this analysis, since calibration errors are indistinguishable from other measurement and recording errors.)

General Approach

The approach used in this study was to define an unfolded spectrum using a series of basis functions:

$$S_u(E)=\sum_{j=1}^{nj} a_j F_j(E,P) \tag{2}$$

where

$S_u(E)$ = the unfolded spectrum
a_j = the coefficient for the jth basis function
$F_j(E,P)$ = the jth basis function
nj = the number of basis functions
E = energy
P = the parameter(s) of F_j

The next step was to compute a χ^2 test statistic using the ratios of the ideal signal to the predicted signal and the ratios of the unfolded signal to the predicted signal, together with the measurement uncertainties of these ratios. These ratios are defined in detail in Section III. This χ^2 test statistic was then minimized using the Fletcher-Powell technique to vary the parameters and weighting factors or coefficients in the basis functions of the unfolded spectrum. The term parameters will refer to the parameters of the function as well as the coefficients for the function in the remainder of this study.

Sequence Of Presentation

Section II of this study presents a detailed analysis of the problem including other deconvolution methods currently being used and possible errors to be considered. Section III presents the theory used to develop the computer program while Section IV presents the development of the computer program. Section V presents the results and a discussion of these results for the validation cases and the test cases conducted using the computer program. The test cases investigated the strengths and weaknesses of the two types of basis functions and the effect simulated measurement error in the measured-to-predicted ratios had on the unfolding technique using these basis functions. Section VI then summarizes the study and presents my conclusions. Recommendations for future studies are presented in Section VII.

II. Detailed Problem Analysis

Introduction

In Section I, the main problem addressed was stated to be the determination of the methodology and the implementation of a computer code to approximate the actual spectrum in Equation (1), the Fredholm integral equation. The purpose of this section is to describe the types of errors found in unfolding techniques, to explain how these errors were handled, and to present the two most popular unfold techniques.

Errors Introduced During Unfolding

During the unfolding process, four main errors contribute to the uncertainty in the unfolded spectrum. The first is simply the measurement error introduced in the measured signals. If the same test was conducted a number of times, a mean value for the measured signals could be established along with a standard deviation. This measurement error includes the errors introduced form the transmission of the ideal signals, the recording of the ideal signals, and the reading of the ideal signals. However, in underground testing and the detection of cosmic radiation by satellites, the experimenter only makes one test. Thus, the experimenter must be aware of and account for the possible error introduced by the statistical nature of the measured signals.

The second error is found in the response functions of the detectors. No matter how carefully an experimenter calibrates the detectors, one will not be able to determine the exact response functions of the detectors. This difference between the exact response functions and the calibrated response functions is the second type of error introduced in the unfolding procedure.

The third error is found in the mathematics of the unfolding process. The error is introduced by converting the Fredholm integral into a summation over very small energy intervals. However, this error can be reduced so as to be negligible compared to the other errors by selecting the proper energy intervals or bin widths to evaluate the integral.

The final error to be discussed is that caused by the nonuniqueness of the solution or approximation to the actual spectrum. Because the approximation of the actual spectrum requires an infinite amount of detail to be resolved from a finite amount of information, the solution is non-unique and the problem is ill-posed. The limited number of detectors and their response functions ensure that the experimenter can not resolve all the detail of the spectrum. Thus, the unfolded signals of a number of spectra could be identical even though the spectra are different. If a detector with a response function which is shaped like a rectangle with neglible width was available at every energy level, the actual spectrum could be determined exactly. Since this is not possible or feasible, the experimenter must be reminded of the nonuniqueness of the approximation to actual spectrum or the unfolded spectrum.

Errors in Actual Practice Versus Simulated Errors in This Study

Of the four errors introduced above, two are handled differently in this study than they are in actual practice. In actual practice, the measurement error and the error introduced by not knowing the exact response functions are present in the measured signals. These errors are then carried over to the measured-to-predicted ratios. However in order to account for these errors, the experimenter will approximate the standard deviation of these measured-to-predicted ratios, σ_i, and use this as a weighting factor for the χ^2 test statistics as discussed in the next section. This σ_i is based on the knowledge gained from past experiments and the calibrations conducted on the detectors.

In this study, it was assumed that the exact response functions were equal to the calibrated response functions and since the signals were not measured, recorde, or transmitted, the measurement errors normally found in the measured signals were not present. Thus, when an actual spectrum was folded with the calibrated response functions to determine a set of measured signals, these two types of error, the measurement error and the calibration error, were not present and the measured signal was equal to the ideal signal. In order to introduce these errors in this study, an option to apply a normal distribution to the measured-to-predicted ratios was included. This is discussed in detail in Section III.

The third error, the error due to evaluating the Fredholm integrals, is present in both the actual case and this study. As discussed in Section III, the error becomes negligible in both cases by selecting a small bin width. Finally the fourth error is also a part of both cases and was demonstrated by studying the unfolding of an actual spectrum using the two different sets of basis functions.

The mathematical propagation of the four errors discussed above was not considered in this study. However, the propagation of the errors was demonstrated by the study conducted using the simulated measurement error. Also, the calibration error may be considered as part of the simulated measurement error.

Current Unfolding Techniques

A number of computer programs have been developed to approximate the actual spectrum implicitly given in Equation (1). However, two techniques, the iterative technique and the cubic spline technique, are the most common (6). This section will describe the two techniques and some of their weaknesses.

Iterative Technique (6). The widely used iterative technique starts by folding a trial spectrum with the calibrated response functions of the detectors and comparing these measured signals to the measured signals from the experiment. The trial spectrum is then modified and smoothed to ensure non-negativity and continuity. The procedure is then repeated until the two sets of measured signals converge. By allowing the measured signals to converge, various artifacts such as discontinuities and spikes are usually introduced in the trial spectrum. Thus, the procedure is generally continued only until the computed signals for the trial spectrum are brought into an acceptable agreement with the measured signals, but stopped before unacceptable artifacts develop in the trial spectrum.

The iterative technique has three main limitations. First, because the iteration is not allowed to converge, the shape of the unfolded spectrum depends on the initial guess at the trial spectrum. Secondly, the technique provides little information to allow for error analysis. Finally, the decision to stop the iteration before unacceptable artifacts develop forces the technique to be user dependent.

Cubic Spline Technique (6). The cubic spline technique used here is based on the method developed at the Lockheed Palo Alto Research Laboratory. The technique consists of building an unfolded spectrum from a set of cubic splines with each cubic spline in the set being multiplied by an expansion coefficient. The cubic splines used are piecewise Lagrangian interpolating splines and are not the common B-splines. In the cubic spline technique the knot locations are adjusted and the number of knots used is varied to obtain the best unfolded spectrum or approximation to the actual spectrum. The cubic spline technique is an excellent approximating technique; however, the main limitation of the cubic spline technique is the dependence on the knot locations or points used to develop the cubic splines. This limitation is reduced by selecting the best locations for the knots before developing the cubic splines.

III. Theoretical Development

Introduction

In the last section, the overall problem and the two most common techniques to solve this problem were presented. This section presents the theory used to develop the methodology for the computer program in this study, including the selection of the response functions; the definition of the various spectra and signals; the formulation of a χ^2 test statistic; the minimization of this χ^2 test statistic; the calculation of a normal distribution for simulating measurement error; and the inclusion of a flux non-negativity constraint.

Response Functions

As stated in Section I, this study assumes the exact response functions are equal to the calibrated response functions and that twenty detectors will be used in the experiment. Two types of detector systems will be used in this study. The first detection system is a filter-fluorescer detection system and produces a closed response function. This closed response function consists of a section that is similar to a narrow rectangle and is an approximation to an ideal response function. This section is referred to as the "inband" response and has a response between the k-edge of the fluorescer and the k-edge of the filter. However, a second section or tail section is also present in the closed response function. The tail section is formed by the response of the detector to energies above the k-edge of the filter. The closed response function is depicted in Figure 1.

The second detection system is a fluorescer detection system and produces an open response function. This second set of response functions allows the experimenter to determine the measured signals using a different set of response functions and thus reduce the possibility of errors in the measured signals from the

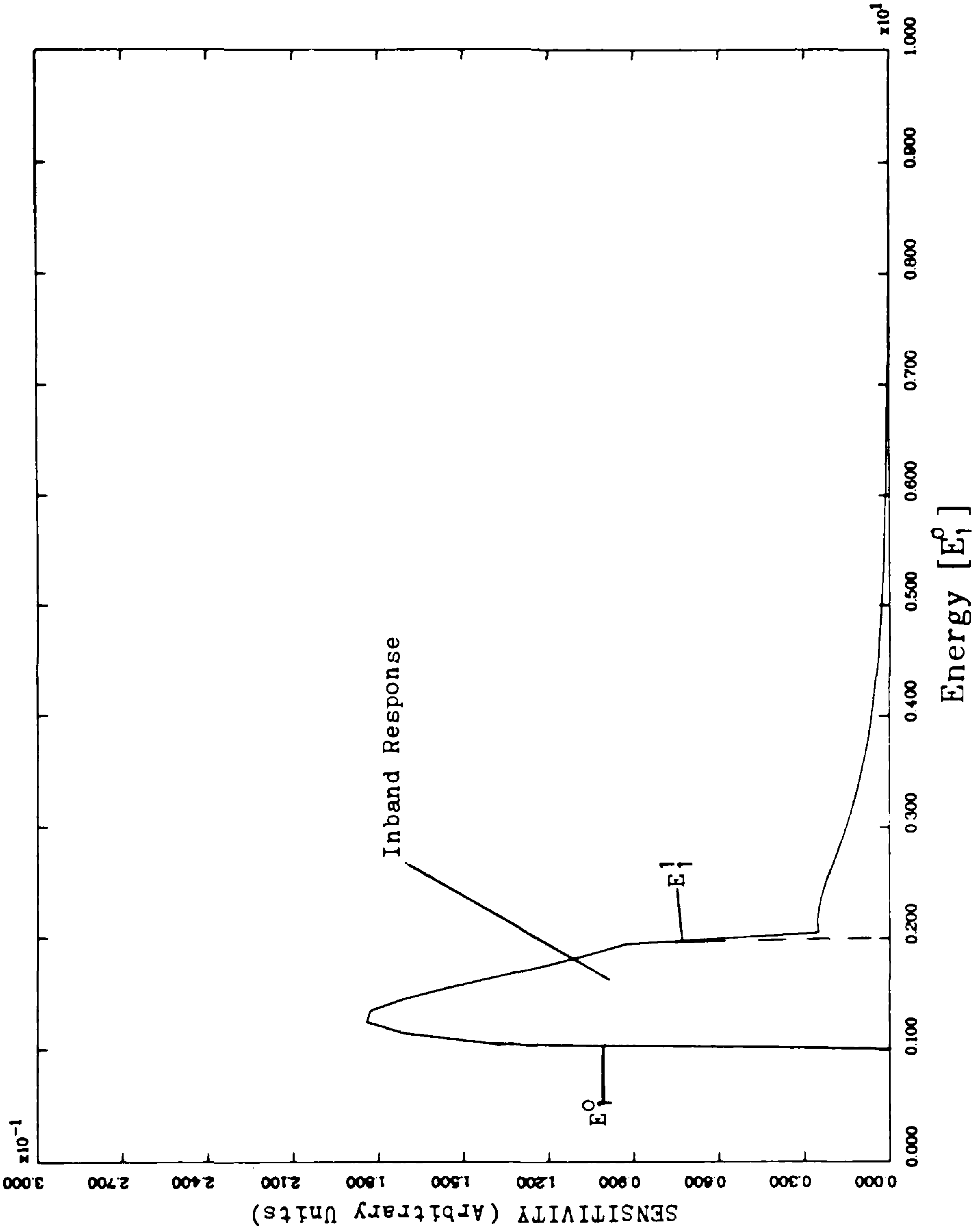


Figure 1. Linear Plot of the Closed Response Function

response functions. The inband response function for this system is defined to be of the same width as the corresponding closed response function. This response function is depicted in Figure 2.

The next step was to define the open and closed response functions. The derivation of the response functions can be found in Appendix A, and is based on a simplification of the detector response function presented by G.M. Gorbachenko and others in reference (4). The response functions used in the study are asymmetric and present a simple, but realistic and analytically representable shape. Additionally, the response functions allow for an unbiased unfolding using various basis functions to define the unfolded spectrum since ; the response functions are not modeled by planckians or cubic splines. Equations (3) and (4) represent the closed and open response functions used in this study.

$$R_i^c(E)=\begin{cases} 0.0 & E < E_i^0 \\ \left(\frac{1}{E}\right)\left\{1-\exp\left[-2.0\left(\frac{E_i^0}{E}\right)^{\beta}\right]\right\}\left\{\exp\left[-0.25\left(\frac{E_i^1}{E}\right)\right]\right\} & E_i^0 \leq E < E_i^1 \\ \left(\frac{1}{E}\right)\left\{1-\exp\left[-2.0\left(\frac{E_i^0}{E}\right)^{\beta}\right]\right\}\left\{\exp\left[-1.5\left(\frac{E_i^1}{E}\right)^{\beta}\right]\right\} & E \geq E_i^1 \end{cases} \tag{3}$$

$$R_i^o(E) = \begin{cases} 0.0 & E < E_i^0 \\ \left(\frac{1}{E}\right)\left\{1-\exp\left[-3.0\left(\frac{E_i^0}{E}\right)^{\beta}\right]\right\} & E \geq E_i^1 \end{cases} \tag{4}$$

where

R_i^c = the closed function of the ith detector
R_i^o = the open response function of the ith detector
E_i^0 = the k–edge of the fluorescer for the ith detector
E_i^1 = the k–edge of the filter for the ith detector
E = energy
i = the ith detector

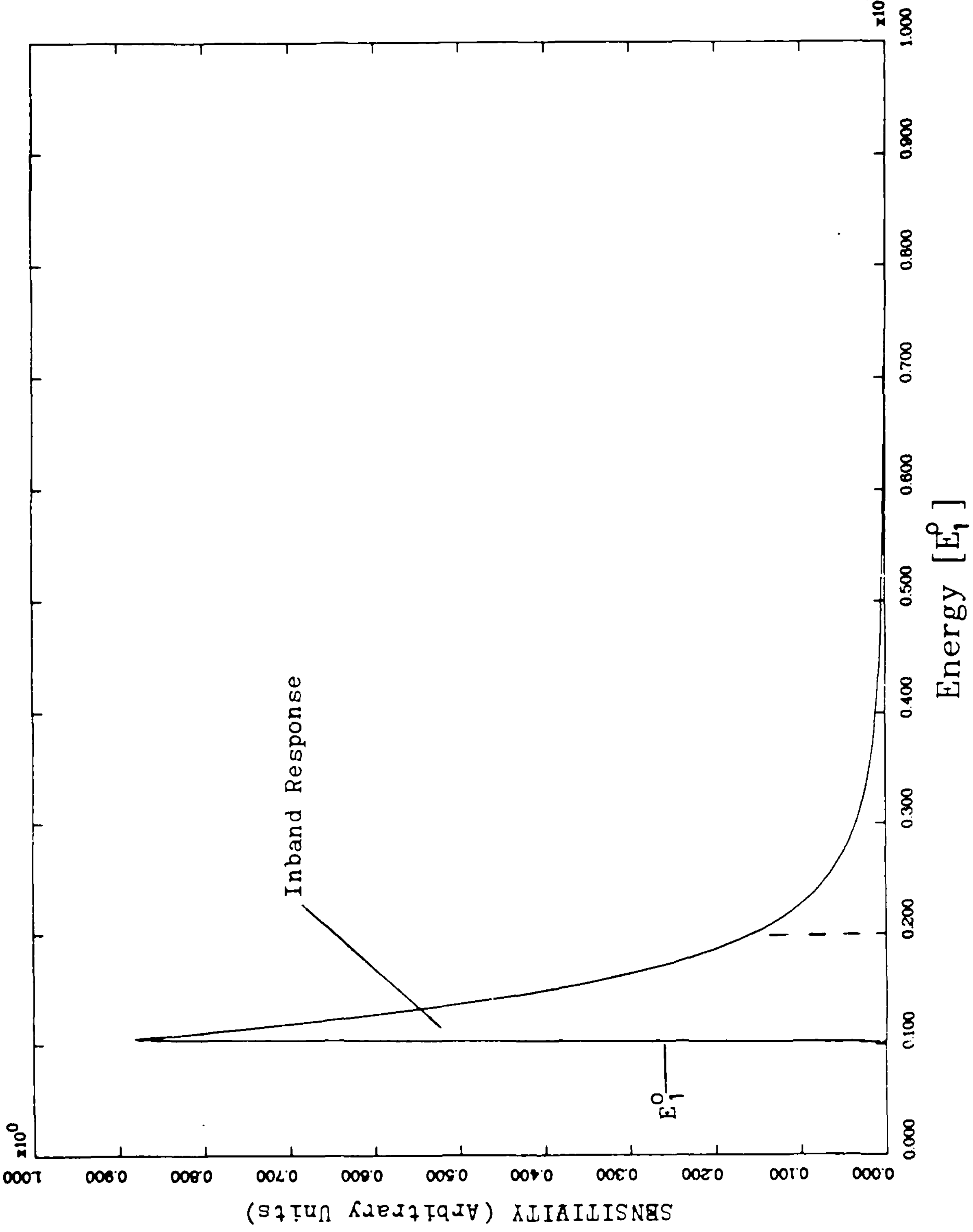


Figure 2. Linear Plot of the Open Response Function

With each detector having a resolving power of about 1.5, the experimenter can achieve a fifty percent overlap in the inband response functions of the detectors by using thirteen closed response detectors. The other seven detectors can then be used as open response detectors covering the entire energy range. With the detectors arranged in this manner, the experimenter is able to detect the entire spectrum using the inband response functions even if a detector fails. This fact is extremely important when fielding experiments. For this detector arrangement, the k-edges of the filter and/or fluorescer for the detectors are as listed in Table I. Thus, Equation (3) corresponds to detectors 1 to 13 and Equation (4) corresponds to detectors 14 to 20.

TABLE I

K-Edges for the 20 Detectors (a)

Det.#	E_i^0	E_i^1	Det.#	E_i^0	E_i^1
1	1.0	2.0	11	12.0	24.0
2	2.0	4.0	12	24.0	48.0
3	4.0	8.0	13	48.0	96.0
4	8.0	16.0	14	1.0	
5	16.0	32.0	15	2.0	
6	32.0	64.0	16	4.0	
7	64.0	128.0	17	8.0	
8	1.5	3.0	18	16.0	
9	3.0	6.0	19	32.0	
10	6.0	12.0	20	64.0	

a. E_i^0 and E_i^1 are computed with respect to E_i^0

Definition of Spectra

During the radiation detection and deconvolution process, three spectra are required. The first spectrum is the actual spectrum ($S_a(E)$). The actual spectrum is unknown and the goal of the deconvolution process is to recover an approximation to the actual spectrum from the measured signals as discussed below. Secondly, the process requires a predicted spectrum ($S_p(E)$). This predicted spectrum is based on basic physics and on all the available knowledge concerning the spectral shape. This predicted spectrum is used to calculate the measured-to-predicted and unfolded-to-predicted ratios. Thus, a wrong predicted spectrum will still allow a proper deconvolution of the approximation to the actual spectrum; but in the iterative scheme and the multiplier spline approach, the detail of the predicted spectrum is intentionally retained (6). Also, an error could be introduced by dividing a very small predicted signal, or a predicted signal of 0.0, into a large measured signal or vice versa. So the best possible predicted spectrum is desired.

The final spectrum required is the unfolded spectrum ($S_u(E)$). This spectrum is the approximation to the actual spectrum and is produced during the deconvolution process (i.e. the unfolding of the Fredholm equation based on the actual spectrum as implicitly defined in Equation (1)). This unfolded spectrum is given by:

$$S_u(E) = \sum_{j=1}^{nj} a_j F_j(E,P) \tag{5}$$

where

nj = the number of basis functions be used
a_j = the coefficient for the jth basis function
$F_j(E,P)$ = the jth basis function and is a function of energy and some other parameter(s) P

The basis functions are the building blocks used to construct the unfolded spectrum. This study examined planckians and cubic splines as sets of basis functions. The planckian function is given by Equation (6) (1:5.5) and the cubic spline basis functions are four point Lagrangian interpolating splines. The splines are discussed in the next section and are derived in Appendix B.

$$F_j(E,P) = \left[\frac{15}{\left(\pi k T\right)^4}\right]\left\{\frac{E^3}{\left[\exp\left(\frac{E}{kT}\right)-1\right]}\right\} \tag{6}$$

where

T = the temperature of the black body distribution
k = the Boltzman constant

Flux Non-Negativity Constraint

In the detection of radiation spectra, a negative radiation intensity at any energy level is physically meaningless. However by choosing an inappropriate set of basis functions for the unfolded spectrum or having faulty data, a negative spectral value may be produced during deconvolution. In order to ensure a positive spectrum, a non-negativity option was included in the computer program. Equation (7) represents the technique used to ensure the spectra remained positive.

$$S_u'(E) = \begin{cases} 0.0 & S_u(E) < 0.0 \\ S_u(E) & S_u(E) \geq 0.0 \end{cases} \qquad (7)$$

where $S_u'(E)$ is the constrained unfolded spectrum

Definition of Signals

Using the three spectra defined previously, four signals are defined. The first is the ideal signal (Y_i^a(E). However, as discussed earlier, the ideal signal is unknown due to calibration error in the response functions and measurement and detection uncertainty. These ideal signals are approximated by the measured signals from the detectors used during the experiment (Y_i^m(E)). These measured signals contain the errors and uncertainty discussed above. This error and uncertainty is represented by σ_i, the estimated standard deviation of the measured-to-predicted ratios caused by the error distribution in Y_i^m. The predicted signal ($Y_i^p(E)$) is then given by:

$$Y_i^p(E) = \int_0^\infty S_p(E) R_i(E) dE \qquad (8)$$

where

$R_i(E)$ = the calibrated response function of the ith detector, either open or closed depending on the detector

The final signal is the unfolded signal, ($Y_i^u(E)$), and is given by:

$$Y_i^u(E) = \int_0^\infty S_u(E) R_i(E) dE \tag{9}$$

Formulation of χ^2 Test Statistic

For convenience, the measurements in this study are specified as ratios to the predicted signals rather than in engineering units (which could not be interpreted without detailed knowledge of the experiment). Thus, two ratios are defined. The first is b_i, the measured-to-predicted ratio. This ratio approximates the ideal-to-predicted ratio. The measured-to-predicted ratio is given by:

$$b_i = \frac{Y_i^m}{Y_i^p} \tag{10}$$

The second ratio is the unfolded-to-predicted ratio, c_i, and is given by:

$$c_i = \frac{Y_i^u}{Y_i^p} \tag{11}$$

Thus, the objective of the unfolding or deconvolution process is to choose the a_j's and parameter(s), P, in the F_j(E,P)'s of Equation (5) to minimize the difference between c_i (which depends on them) and b_i (which is dependent on the measurement data). In order to give appropriately increased weight to the more accurate detectors, a χ^2 test statistic was formulated and minimized. Equation (12) defines the χ^2 test statistic.

$$\chi^2 = \sum_{i=1}^{ni} \left[\frac{\left(c_i - b_i \right)}{\sigma_i} \right]^2 \tag{12}$$

where

ni = the number of detectors
σ_i = the standard deviation in the ith measured–to–predicted ratio

However, before χ^2 can be minimized, the above mentioned signal integrals must be evaluated. In this study, the composite midpoint rule was used to numerically evaluate all integrals. Since the detectors only detected the spectrum above E_1^0, the integrals were evaluated beginning at this energy. Also, the integral was truncated at $128{*}E_1^0$. The approximation to the signal integrals is given in Equation (13).

$$\int_{E_1^0}^{128E_1^0} S_u(E)R_i(E)dE \approx \sum_{k=1}^{nk} S_u(E_k)R_i(E_k)\Delta E_k \tag{13}$$

where

nk = the number of energy bins
E_k = the midpoint energy for the kth energy bin
ΔE_k = the energy bin width of the kth bin

By selecting narrow energy bin widths, the error introduced in the unfolding process because of Equation (13) becomes negligible when compared to the other errors. Table II presents a study using one planckian basis function with a_j equal to 2.0 and a temperature of $5.0E_1^0$ to evaluate the unfolded signal. Due to the equal resolution of the detectors, only one detector was used as an example.

Minimization of χ^2 Test Statistic

Once χ^2 was defined using a trial spectrum, an iterative process to minimize χ^2 was required. Three methods were studied. The first method conducted the minimization by a least squares analysis. This method was useful for linear optimization; but, the method was not applicable for adjusting the temperatures in the planckians or adjusting the knots for the cubic splines. Also, the

TABLE II

Comparison of the Unfolded Signals Using Varying Energy Bin Widths

Bin Width	Y_i^*	R_i
0.05	1.48	closed
	3.65	open
0.10	1.48	closed
	3.65	open
0.25	1.48	closed
	3.65	open
0.50	1.50	closed
	3.66	open
1.00	1.60	closed
	3.66	open

method could not be used with the flux non-negativity constraint. Finally, a large amount of computer time would be used calculating inverse matrices. The calculation of the inverse of a matrix also presented a possible problem. The matrix could be singular or near singular. However, this last problem could be corrected by selecting the appropriate basis functions. Based on the need for a flux non-negativity constraint, the method was rejected.

The second technique studied was the steepest descent method of minimization. This technique proved reliable but required a large number of iterations when the number of parameters in the basis function or the number of basis functions was increased.

The Fletcher-Powell minimization technique was the third method tested and was selected for use in this study. The method consists of calculating the gradient of the function, χ^2, and then multiplying this gradient matrix by a correction matrix which modifies the search direction. A detailed description of the method can be found in references (3) and (5:75-76).

Two problems do exist with this method. The Fletcher-Powell method was designed for those functions for which the gradient can be determined analytically. However in this study, the gradient must be calculated numerically. Thus, an additional error is added into the calculation. Secondly, the method is used to find a local minimum and does not guarantee this local minimum to be the global minimum. However, during this study the local minimum problem was not encountered since the basis functions used contained only one local minimum or global minimum.

Once the search direction was determined using the Fletcher-Powell method, the distance to move in that direction had to be calculated. This distance was calculated by using a line search routine to minimize χ^2(param.+t*s). In this study, t is the step size or distance to move in the search direction and s is the unit search direction calculated in the Fletcher-Powell minimization technique. Note that a search direction is based on the gradient of χ^2 and is a function of all the parameters used in the basis functions.

Two line search routines were evaluated during this study. The first method consisted of calculating the value of χ^2 at two locations and comparing the values. The lower value for χ^2 was retained and the search interval was expanded or reduced in order to further minimize χ^2. This method was continued until the interval between two locations was less than a given value. Finally, the total distance travelled was calculated. This value was then used as the distance to travel in the search direction.

The second method evaluated and the one selected for this study used the value and slope of χ^2 at two locations to construct a cubic fit. This cubic fit was then used to determine the distance to travel in the search direction. Reference (5:76-80) presents the method in detail. One modification to the method was required. If the functional values of χ^2 are equal and the slope of the second or new location in the search direction is 0, then the search distance is set equal to the distance traveled between the original χ^2 location and this final χ^2 location.

Normal Distribution for Simulating Measurement Error

Once the basic unfolding technique was determined, the ability to add simulated measurement error to b_i was added. The method employed to calculate this normal distribution for the simulated measurement error is found in reference (7:949-953) and is given by:

$$Z_i = \sigma_i \left[\left(\sum_{j=1}^{n} U_j - \frac{n}{2} \right) \left(\frac{n}{12} \right)^{0.5} \right] \qquad (14)$$

where

U_j = the jth number in a sequence of psuedo—random numbers distributed in the interval [0,1]
n = the number of psuedo—random numbers utilized

For simplicity, twelve values of U_j were used. These values were obtained using a random number generator available on the UNIX computer system. A sample of ten normal variates generated in this way was tested for skewness and kurtosis. In both tests, the distribution could not be rejected as normal.

IV. Development of the Computer Program

Introduction

In Section III, the basic theory used to develop the methodology involved in the deconvolution or unfolding technique was discussed. This section presents the procedure used to convert this methodology into a general computer program. The program was developed in five basic steps: development of the test spectra; development of a planckian unfolding program; addition of simulated measurement error; development of the cubic spline basis functions; and formalization of the final program. Each step is presented in this section along with a discussion of the problems encountered.

Development of Test Spectra

In the field experiment the signals are produced and measured during the experiment; thus the predicted spectrum, $S_p(E)$, the calibrated response functions, $R_i(E)$, the measured-to-predicted ratios, b_i, and the estimated standard deviations of the measured-to-predicted ratios, σ_i, are known. However in order to conduct this study these values had to be simulated. By selecting parameters for a set of basis functions, an actual spectrum was simulated using the same method as used to produce an unfolded spectrum in Equation (5). The actual spectrum was then folded with the calibrated response functions derived in Section III to simulate the measured signals. The predicted spectrum and signals were also produced in this manner. Finally, the estimated standard deviations for the measured-to-predicted ratios were defined. This program assumes the same standard deviation for each detector.

Planckian Unfolding Code

The second step of the computer development consisted of writing a program based on the methodology in Section III that could unfold the actual spectrum using the same type of basis function for the unfolded spectrum as was used to construct the actual spectrum. In order to simplify the calculation required, the integral used to calculate the predicted signal in Equation (8) was normalized. The calibrated response functions were then multiplied by this normalization constant. Thus, the measured-to-predicted and unfolded-to-predicted ratios are simply the measured signal and unfolded signal. However, the calibrated response functions that were multiplied by the normalization constant must be used to calculate these signals.

To accomplish this, a set of planckian basis functions were selected as the basis functions for the actual, predicted, and unfolded spectra. The computer program used the Fletcher-Powell minimization technique to vary the parameters and coefficients in the basis functions of Equation (5) and determine the best parameters for the unfolded spectrum. The program was validated using one, two, and three basis functions. In all cases, the predicted spectrum was set equal to the actual spectrum.

This validation is discussed in Section V. During this validation, the two methods for determining the search distance is discussed in Section III were compared. Since the cubic fit was faster and required fewer iterations of the line search subroutine, it was selected.

During this validation a possible problem was uncovered even though the problem did not affect the validation. The operator of the computer program should note that it is possible for the delta used in the calculation of the gradient during the minimization of χ^2 to be greater than the step size or distance traveled in the search direction. When this occurs, a possible error in the calculation of χ^2 could result.

The validation of the Fletcher-Powell method also included a comparison of the number of iterations required before convergence for this method and the steepest descent method. The Fletcher-Powell technique was dramatically more efficient and required fewer iterations, especially with the larger number of parameters.

Based on this first program and its modification, the technique selected to minimize χ^2 was the Fletcher-Powell minimization technique with the modified cubic line search routine. In addition the flux non-negativity constraint discussed in Section III was used in this program.

Simulated Measurement Error

The third step of the program development was the user option to add simulated measurement error to b_i. The simulated measurement error was used to account for the calibration errors and the errors from the measurement, recording, and transmission of the ideal spectrum. This was accomplished by inserting the computer code necessary to calculate Equation (14). The simulated measurement error option was included in the program immediately after b_i was calculated and σ_i was defined.

Cubic Spline Basis Function

Once the general program was developed and validated using the planckian basis functions, another basis function subroutine was added to allow for the actual, predicted, and unfolded spectra to be constructed from either planckians or cubic splines.

The cubic spline basis function consisted of two linear segments, a planckian tail, and a variable number of cubic spline segments. The segments were defined between two consecutive knot locations. The cubic splines used were cubic Lagrangian interpolating splines. The basis functions were then defined as the combination of four segment functions. The details of this cubic spline basis function is presented in Appendix B.

Two forms of this new subroutine were considered and developed. The first form required the knots used in determining the cubic splines to be fixed. This form was validated as shown in Section V. However, in order to effectively minimize χ^2, the knots must be considered variable. In other words, the knots were considered parameters of the cubic splines. The final version of the program used the cubic splines with variable knots as a possible type of basis function.

Decon7.f

Decon7.f is the final version of the deconvolution or unfolding program presented in this study. The computer program allows for planckian basis functions, cubic spline basis functions, or other spectra (input from a file) to be used as the actual and predicted spectra and either the planckians or the cubic splines can be used to define the unfolded spectrum. The program then uses the Fletcher-Powell minimization technique combined with the modified cubic line search routine to minimize χ^2 by modifying the parameters in the basis functions used to produce the final unfolded spectrum. The next section will present the validation and results of this program. In addition, the documentation and pseudo-code for the program are presented in Appendix C and the source code is presented in Appendix D.

V. Results and Discussion

Introduction

Section IV discussed the development of the computer program, during which several validation cases were considered. This section presents those validation cases as well as the results from the preliminary study on the effects of simulated measurement error in the measured to predicted ratios and the weaknesses of the two types of basis functions. For this study a final χ^2 value of less than or equal to the number of detectors used was considered acceptable. Also, in this section, the initial spectrum refers to the initial guess at the unfolded spectrum, and the actual and predicted spectra are identical.

Validation Cases

The validation cases presented were developed to ensure the computer program was functioning properly and to test the methodology used to construct the computer program. The goal of the validation cases was to unfold the exact spectrum that was used to construct the measured signals. For these cases, an arbitrary spectrum was selected as the actual spectrum and this spectrum was then folded with the calibrated response functions to define the measured signals. As mentioned above, the predicted spectrum was identical to the actual spectrum. Also, simulated measurement error was not added to the measured-to-predicted ratios. Finally, another arbitrary spectrum was selected as the initial guess at the unfolded spectrum. In order to validate the computer program and the methodology, the unfolded spectrum should converge to the actual spectrum.

Case BP. These cases represent the benchmark or validation cases for the planckian basis functions. The three cases presented validate the use of one, two, and three basis functions. The results of these validation cases are presented in Table III. In addition, a more detailed validation of the one basis function case is presented in Appendix E. In all the planckian benchmark cases, the unfolded spectrum was indistinguishable from the actual spectrum when plotted as noted in Table III since the parameters of the unfolded spectrum converged to those of the actual spectrum.

TABLE III
Validation Cases for Planckian Basis Functions (a,b)

Case	Spectrum	Parameters(c)						χ^2
		BF1		BF2		BF3		
		a	T	a	T	a	T	
BP1	Actual	2.0	5.0					
	Initial	1.0	6.0					
	Unfolded	2.0	5.0					9.8E-4
BP2	Actual	3.0	6.0	8.0	10.0			
	Initial	5.0	2.0	1.0	7.0			
	Unfolded	3.0	6.0	8.0	10.0			5.9E-3
BP3	Actual	8.0	5.0	2.0	10.0	1.0	15.0	
	Initial	3.0	2.0	4.0	6.0	2.0	8.0	
	Unfolded	8.0	5.0	1.9	9.9	1.1	15.0	2.6E-2

a. Convergence criteria: $\chi^2 \leq 0.01$ or less than a 1% change in χ^2 for successive iterations

b. $\sigma_i = .01$

c. BF stands for basis function

Case BF. The first four BF cases represent using cubic splines with fixed knots as the basis function and using the same knots for the actual, predicted, and unfolded spectra. These cases validated the use of the cubic spline basis functions with the methodology validated in the BP cases. The results of these cases are presented in Table IV. Once again, the unfolded and actual spectra were identical when plotted and the parameters of the spectra also converged.

Four additional cases were also considered. These last four cases were used to validate the applicability of using fixed knots. In these cases, the knots used to form the actual and predicted spectra were different than those used to form the unfolded spectra. As expected, the deconvolution technique was unable to correct for the error in the knot locations.

Thus, all four cases yielded a χ^2 of greater than 20. The table of results and plots for these cases can be found in Appendix E. Based on these results, the knot locations were included as parameters in the basis functions.

TABLE IV
Validation Cases for Cubic Spline
Basis Functions with Fixed Knots (a,b,c)

Case	Spectrum	Cubic Coefficient					T	χ^2
BF1 & BF2(d)	Actual	4.0	5.0	3.0	1.6	1.0	10.0	
BF1	Initial	2.0	7.0	1.0	3.0	4.0	10.0	
	Unfolded	4.0	5.0	3.0	1.6	1.0	10.0	4.3E-3
BF2	Initial	2.0	7.0	1.0	3.0	4.0	20.0	
	Unfolded	4.0	5.0	3.0	1.6	1.0	10.0	4.3E-4
BF3 & BF4(e)	Actual	4.0	5.0	3.0	1.6	1.0	10.0	
BF3	Initial	2.0	7.0	1.0	3.0	4.0	20.0	
	Unfolded	4.0	5.0	3.0	1.6	1.0	10.0	2.4E-3
BF4	Initial	2.0	7.0	1.0	3.0	4.0	2.0	
	Unfolded	4.0	5.0	3.0	1.6	1.0	10.0	6.6E-4

a. Convergence criteria: $\chi^2 \leq 0.01$ or less than a 0.01% change in χ^2 for successive iterations.

b. $\sigma_i = 0.01$

c. Fixed knots at 0.0,1.0,128.0 are implicit in definition of these splines and are not listed.

d. Knots for case BF1 and BF2 were 5.0,10.0,25.0,50.0

e. Knots for case BF3 and BF4 were 10.0,25.0,50.0,80.0

Case BC. The BC cases were formulated to validate the computer program using the cubic spline basis functions with variable knots. As explained in Appendix B, these basis functions were formed by combining four, four point Lagrangian interpolating polynomials. Each basis function was a function of the knots selected to construct the polynomials and the intensity at these knot locations. The results of these cases are presented in Table V. The linear plots of the actual and unfolded spectra for all cases are located in Appendix E.

These validation cases presented the following noteworthy points. First, the study presented the difficulty the computer program had in varying the knot location. The program was only able to determine an acceptable spectra in three of

TABLE V
Validation Cases for Cubic Spline
Basis Functions with Variable Knots (a,b,c)

Case	Spectrum	Parameters					χ^2
		Knots		Coefficients		T	
BC1	Actual	5.0	30.0	5.0	3.0	15.0	
		60.0		2.0	1.0		
	Initial	2.0	15.0	1.0	4.0	8.0	
		35.0		6	0,2.0		
	Unfolded	12.0	13.0	4.8	1.5	13.0	230.0
		28.0		0.44	1.5		
BC2	Actual	5.0	10.0	4.0	5.0	10.0	
		25.0	50.0	3.0	1.6		
				1.0			
	Initial	2.0	15.0	3.0	2.0	5.0	
		35.0	60.0	5.0	1.0		
				2.0			
	Unfolded	4.0	11.0	4.0	4.9	10.0	5.4
		28.0	53.0	2.7	1.6		
				0.88			
BC3	Actual	5.0	10.0	4.0	6.0	10.0	
		25.0	50.0	5.0	3.0		
		80.0		2.0	1.0		
	Initial	2.0	15.0	1.0	3.0	4.0	
		30.0	45.0	5.0	4.0		
		60.0		3.0	2.0		
	Unfolded	2.4	13.0	3.9	4.8	12.0	19.0
		37.0	49.0	4.2	2.3		
		62.0		2.0	1.8		
BC4	Actual	5.0	30.0	5.0	3.0	15.0	
		60.0		2.0	1.0		
	Initial	4.0	35.0	4.0	4.0	13.0	
		50.0		3.0	2.0		
	Unfolded	4.5	30.0	5.0	3.2	15.0	0.026
		60.0		2.0	1.0		

a. $\sigma_i = 0.01$

b. Convergence criteria: $\chi^2 \leq 0.01$ or less than 0.1% change in χ^2 for successive iterations

c. Fixed knots at 0.0,1.0,128.0 are implicit in definition of these splines and are not listed.

the four cases. The case that was rejected used three fixed knots and three variable knots. This case represented the minimum number of knots possible in order to have at least one section of the spectrum that is constructed from four cubic splines. This difficulty is most likely due to the fact that the six knot cubic is constrained to a linear or planckian fit in all but one section. Thus, the small variations in the knot locations cause only a small change in χ^2 and the computer program is finally stopped due to such small changes in χ^2. More computer time could be used to determine if the value for χ^2 is finally reduced but a better solution is to select another set of basis functions, add more knots, or change the knot locations. This is shown by the accuracy of the unfolded spectrum for case BC4 when the initial guess was close to the actual spectrum as compared to case BC1.

Finally, the validation results for the cubic spline basis functions show the need for an accurate guess at the knot locations. By optimizing the selection of knots for the initial guess at the unfolded spectrum, the cubic splines with variable knots would resemble the cubic splines with fixed knots and a better unfolded spectrum should be produced. This optimization could be conducted by unfolding signals produced using the predicted prior to unfolding the real data. However, this will not ensure the best knot are selected since the actual spectrum may not compare to the predicted. Thus, the dependency of the cubic spline deconvolution technique on the knot locations is verified.

Test Cases

Once the final computer program was validated, two preliminary studies were conducted. The first was a study to determine the effect of simulated measurement error in the b_i's on the deconvolution process. The second study was conducted to demonstrate the degree to which the unfolded spectrum would approximate the actual spectrum if different basis functions were used for the respective spectra.

Simulated Measurement Error Study. As stated, this study consisted of adding a simulated measurement error to b_i in order to simulate the measurement errors discussed is Sections II and IV and which are present in the experimental data. This simulated measurement error was constructed using Equation (14). As noted, the normal distribution was scaled by σ_i, the standard deviation of the measured to predicted ratio. Ten cases with simulated measurement error and one benchmark case without the simulated error were tested for each type of basis function. A σ_i of 0.15 was used in all cases. This σ_i was a relative error and not an absolute error. In other words, the σ_i's were the percent of error in the b_i's. The benchmark case was labeled case 0 while all other cases were numbered according to the number used as the seed for the random number generator. Tables VI and VII represent the results of this study. Figures 3 thru 6 depict the combined spectra for each case. Appendix F contains the individual plots of the actual and unfolded spectra for each case.

TABLE VI
Noise Study For
Planckian Basis Functions (a)

Case	Parameters						χ^2	TF(c)
	BF1		BF2		BF3			
	a	T	a	T	a	T		
Actual	8.0	5.0	2.0	10.0	1.0	15.0		11.0
Initial(d)	3.0	2.0	4.0	6.0	2.0	8.0		9.0
NP0	6.4	4.8	2.5	6.6	2.1	13.0	0.010	11.0
NP01	6.1	4.6	3.0	7.5	2.1	11.0	18.0	11.2
NP02	6.2	4.8	2.7	6.3	1.8	15.0	29.0	10.7
NP03	6.5	4.9	2.4	6.8	2.1	13.0	16.0	11.0
NP04	6.6	5.0	2.5	7.2	1.8	13.0	6.0	10.9
NP05	4.8	4.2	3.5	7.0	2.1	12.0	13.0	10.4
NP06	6.1	4.7	2.5	7.0	1.9	13.0	12.0	10.5
NP07	6.4	4.8	3.0	7.0	1.8	13.0	9.7	11.2
NP08	6.0	4.9	2.5	7.3	1.8	13.0	17.0	10.3
NP09	6.2	5.0	3.1	7.0	1.4	13.0	15.0	10.7
NP10	6.1	4.7	2.6	6.9	2.0	13.0	18.0	10.7

a. Convergence criteria: $\chi^2 \leq 0.01$ or less than a 0.1% change in χ^2 for successive iterations

b. BF represents basis function

c. TF stands for the total fluence from 0.0 to ∞

d. Same initial spectrum for all cases, results represent unfolded spectrum

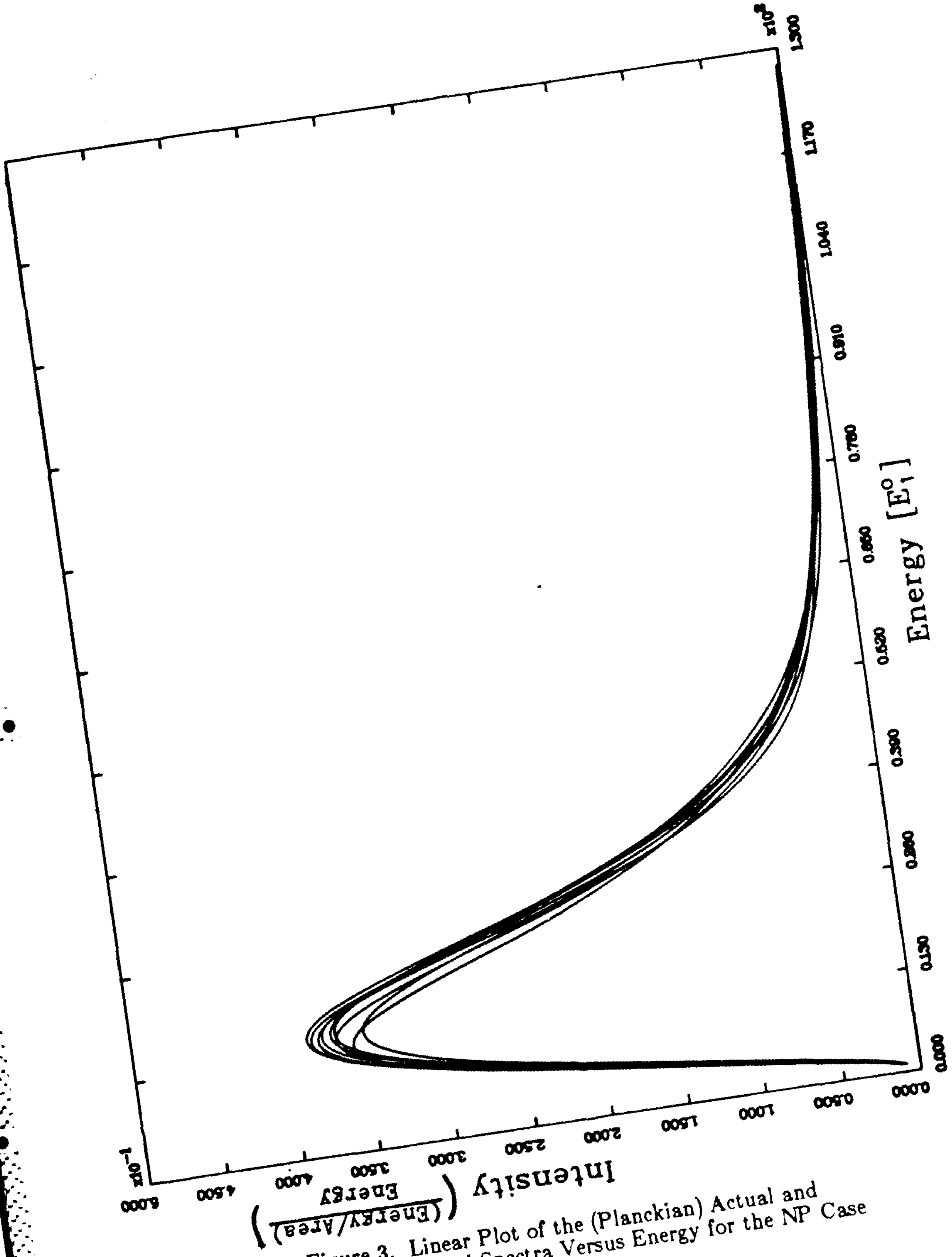


Figure 3. Linear Plot of the (Planckian) Actual and (Planckian) Unfolded Spectra Versus Energy for the NP Case

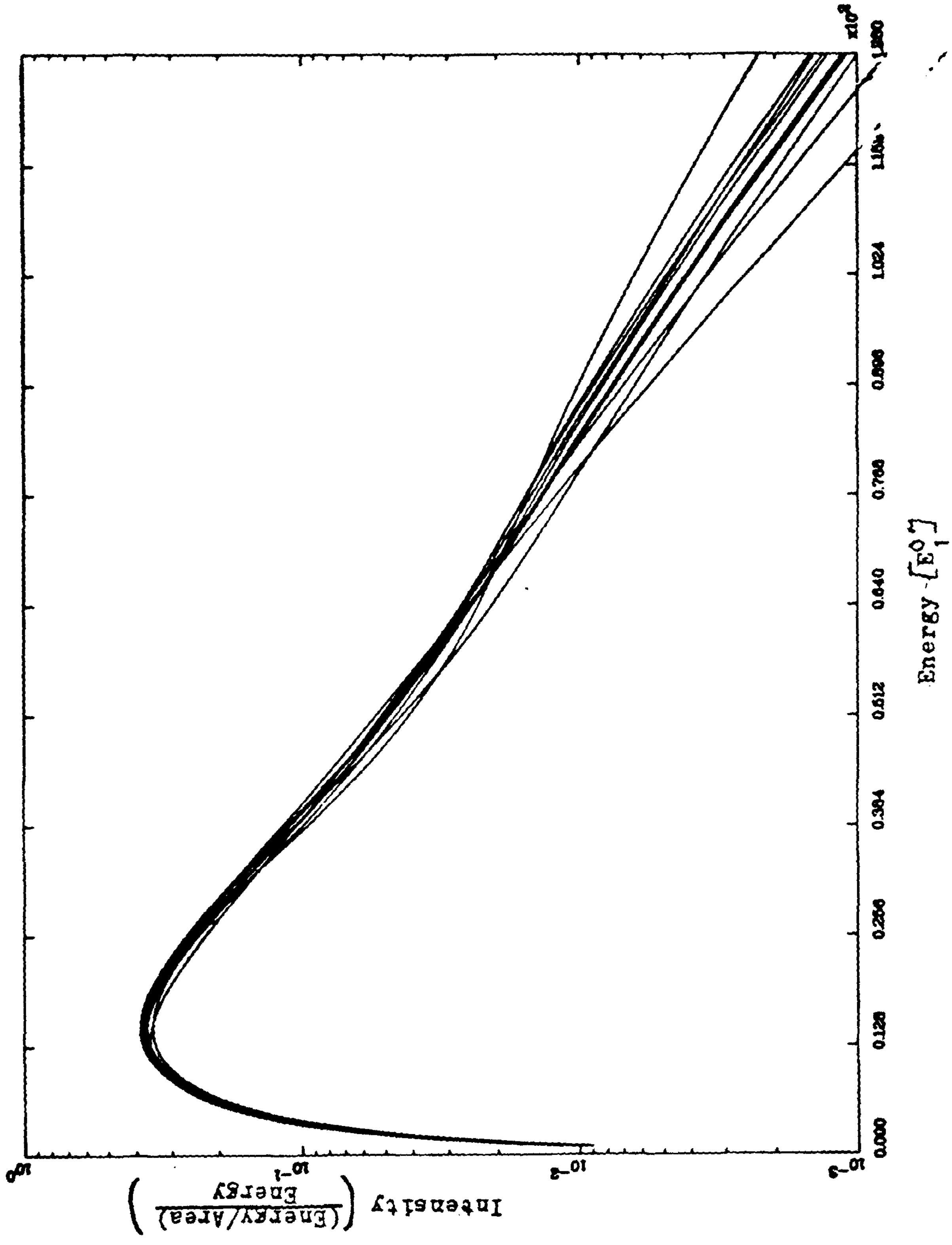


Figure 4. Semi-log Plot of the (Planckian) Actual and (Planckian) Unfolded Spectra Versus Energy for the NP Cases

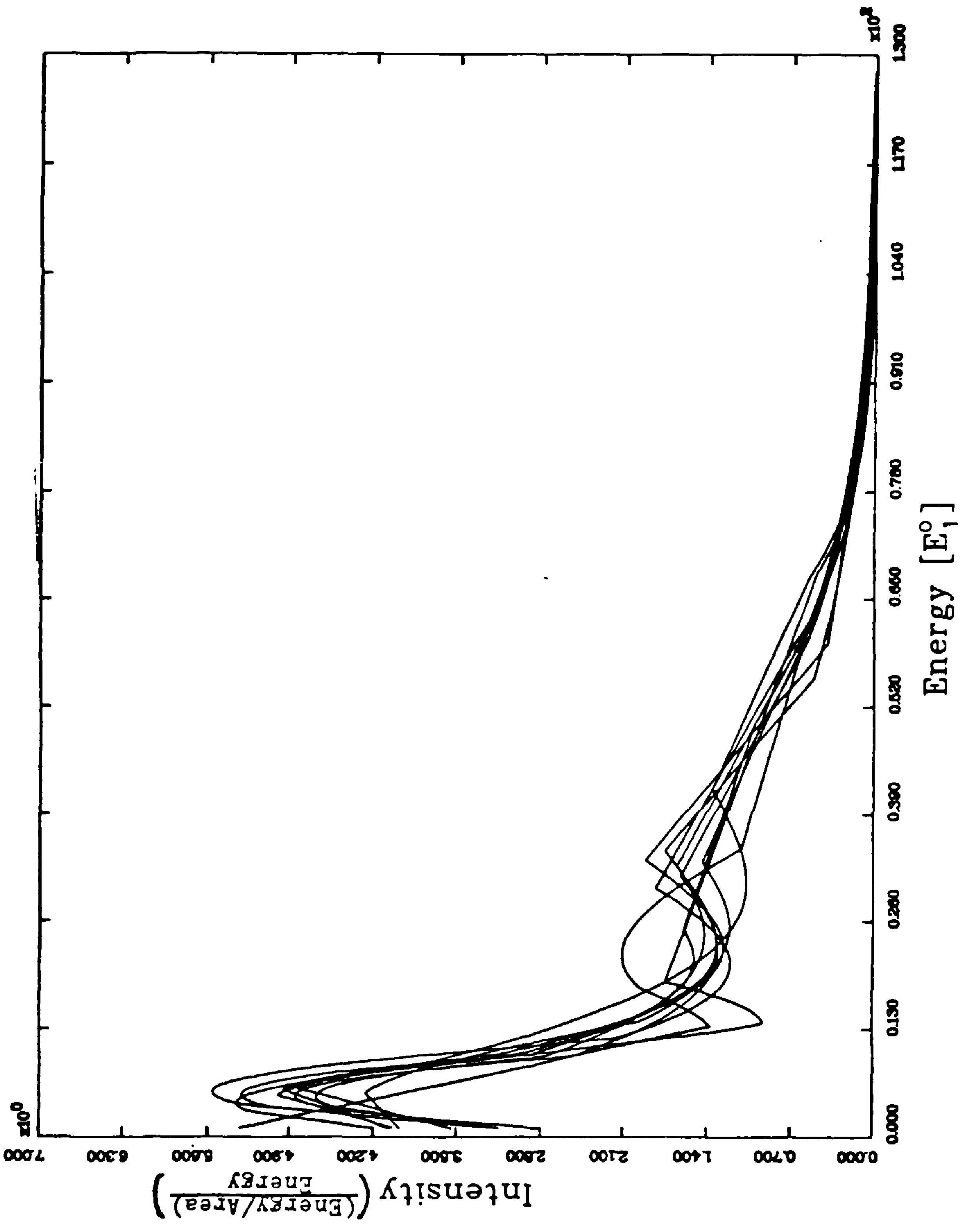


Figure 5. Linear Plot of the (Spline) Actual and (Spline) Unfolded Spectra Versus Energy for the NC Cases

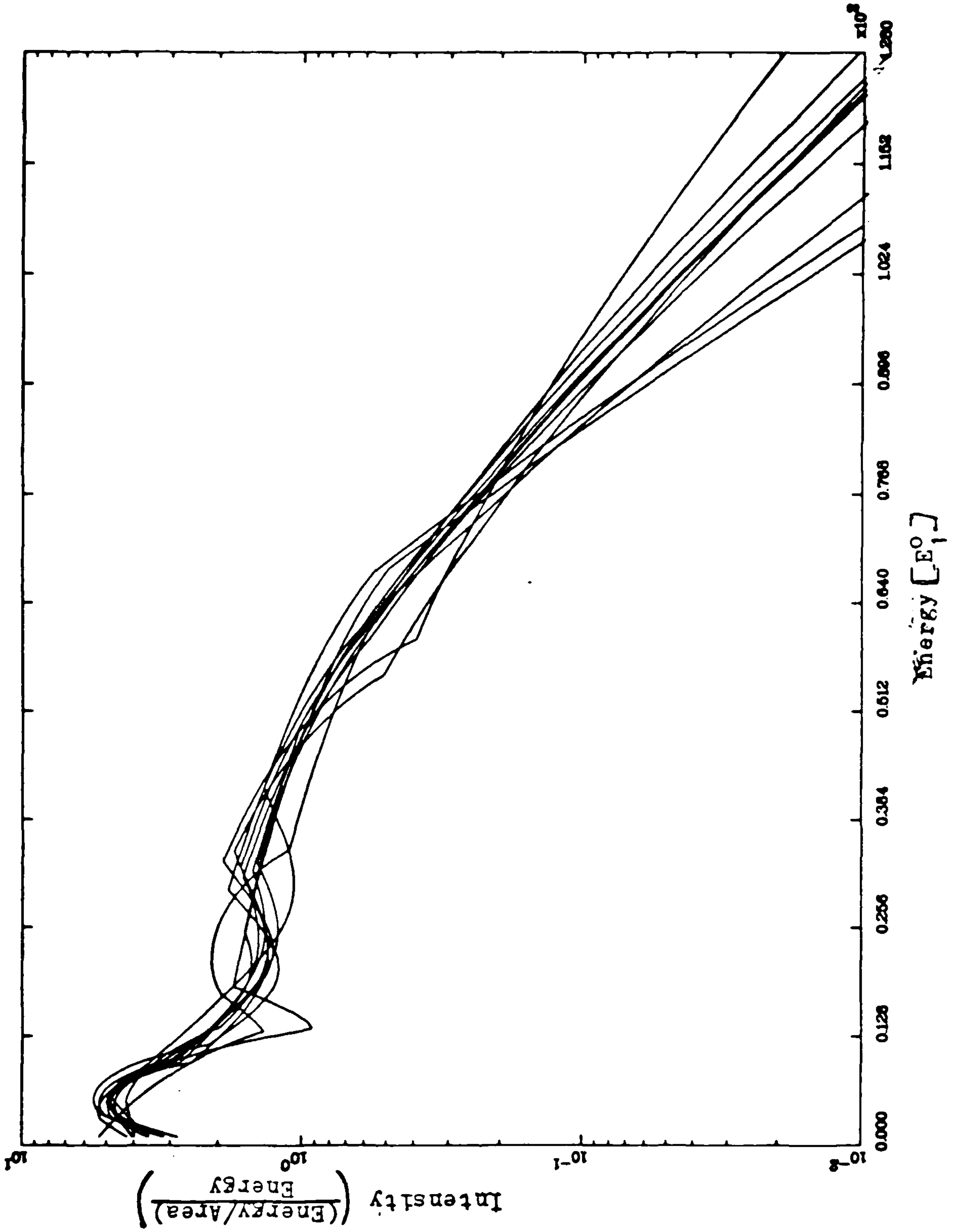


Figure 6. Semi-log Plot of the (Spline) Actual and (Spline) Unfolded Spectra Versus Energy for the NC Cases

TABLE VII

Noise Study for Cubic Spline Basis Function with Variable Knots (a,b)

Case	Parameters					T	χ^2
	Knots		Coefficient				
Actual	5.0	10.0	4.0	5.0	3.0	10.0	
	25.0	50.0	1.6	1.0			
Input(c)	2.0	15.0	3.0	2.0	5.0	5.0	
	35.0	60.0	1.0	2.0			
NC0	4.4	11.0	4.0	4.9	2.7	9.7	0.020
	27.0	60.0	1.6	0.68			
NC01	3.8	12.0	3.1	5.3	2.2	8.5	11.0
	30.0	60.0	1.8	0.66			
NC02	13.0	18.0	5.3	1.4	2.0	11.0	17.0
	35.0	61.0	1.1	0.57			
NC03	5.7	8.7	3.5	4.8	3.8	7.5	8.6
	42.0	68.0	1.4	0.55			
NC04	3.5	9.7	3.1	4.5	2.8	9.6	5.5
	33.0	60.0	1.7	0.62			
NC05	3.1	14.0	4.2	5.3	0.92	7.3	9.4
	19.0	68.0	1.8	0.49			
NC06	3.6	10.0	3.1	5.0	2.9	10.0	8.8
	33.0	60.0	1.4	0.61			
NC07	5.7	10.0	4.1	5.0	3.2	13.0	7.2
	33.0	60.0	1.9	0.39			
NC08	3.2	9.7	2.8	4.5	2.5	10.0	15.0
	31.0	60.0	1.6	0.58			
NC09	5.2	14.0	4.0	4.3	2.0	11.0	13.0
	35.0	56.0	1.8	0.51			
NC10	3.4	9.4	3.6	4.8	3.0	10.0	18.0
	32.0	60.0	1.5	0.66			

a. Convergence criteria: $\chi^2 \leq 0.01$ or less than 0.1% change in χ^2 for successive iterations

b. Fixed knots at 0.0,1.0,128.0 are implicit in definition of these splines and are not listed

c. Same initial spectrum for all cases, results represent the unfolded spectrum

The results of the preliminary study concerning simulated measurement error show that measurement error in the measured signals has little effect on the unfolded spectrum. In all but one case, the final χ^2 was less than the number of instruments used, 20. The relation between the locations or amounts of measurement error and χ^2 was not determined in this study. However, this could be considered in a future study. In addition to this result, the simulated measurement error study on the planckian basis functions showed that the temperatures produced in the unfolded spectrum should not be considered the actual black body temperatures, for multi-temperature spectra. However since only one spectrum was considered a more detailed study should be conducted. This is shown by the temperatures arrived at for the unfolded spectrum during the simulated measurement error study. Also, since the planckians were normalized, the sum of the coefficients approximates the total fluence for the spectrum. It should be noted that the total fluence is accurate to about 5% in the planckian cases tested. The measured-to-predicted ratios are accurate to 15% but some of the ratios are too large while some are too small so the total fluence error is less or the accuracy is better for the total fluence measurement.

It should also be noted that the values for χ^2 were smaller for the cubic spline basis functions than for the planckian basis functions. Because the cubic splines are local functions, the splines can fit the detail of the spectrum better than the global functions like the planckians. Thus, the χ^2 should be lower.

Applicability of the Basis Functions Study. During the unfolding of real data, the actual spectrum is not known so one can not select the set of basis functions that were used to construct the actual spectrum. This study was used to determine how well a set of basis functions could approximate an actual spectrum constructed from a different set of basis functions. Two actual spectra were approximated using an ideal situation in which simulated measurement error was not added to the measured to predicted ratios. Also in both cases the non-negativity constraint was applied. The first was an actual spectrum constructed from three planckian basis functions. This spectrum was then approximated using cubic spline basis functions with six, seven, and eight knots. The results of this study are presented in Table VIII and the best approximation to the actual spectrum is presented in Figures 7 and 8. All plots of the actual and unfolded spectra are presented in Appendix F.

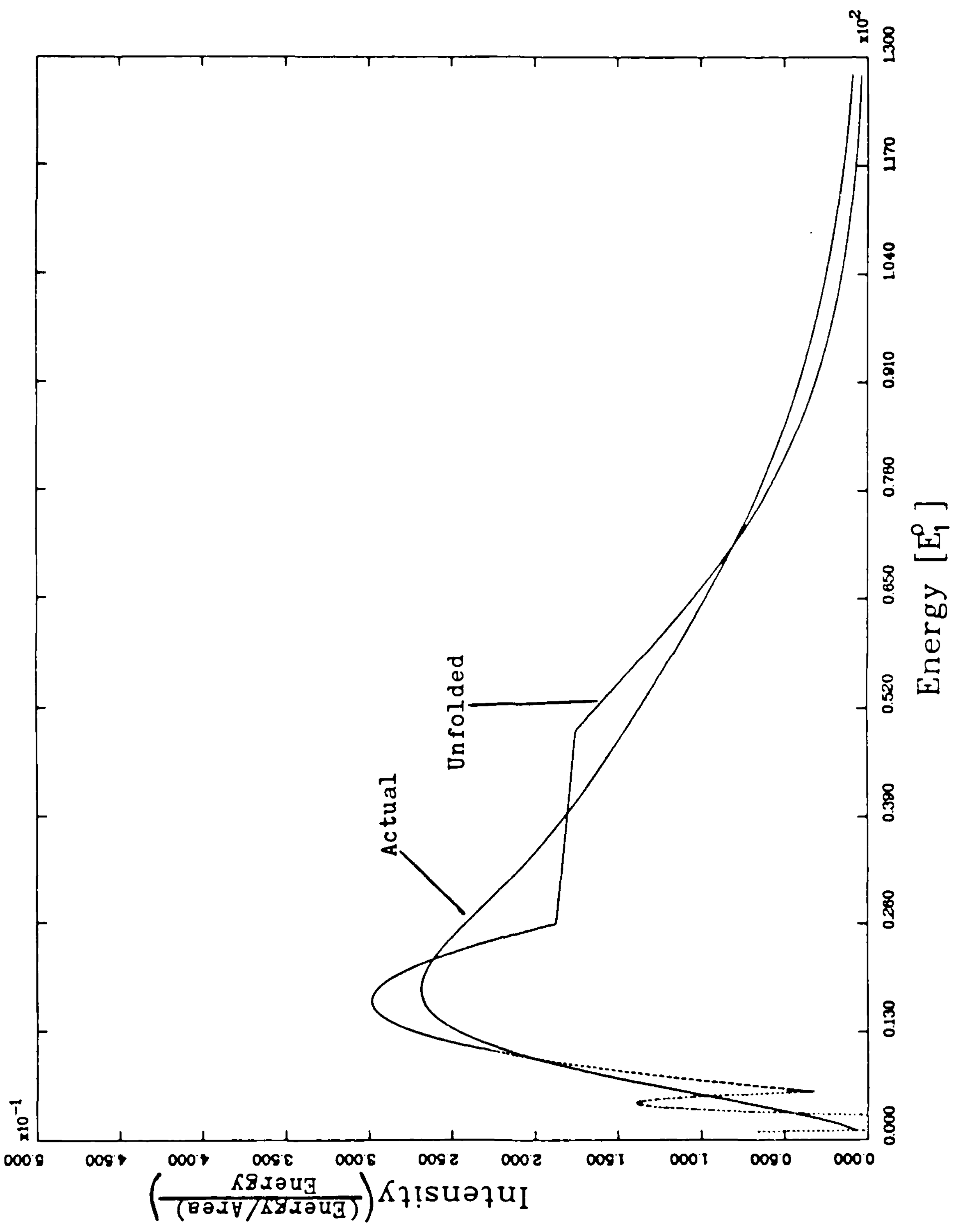


Figure 7. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case PC3

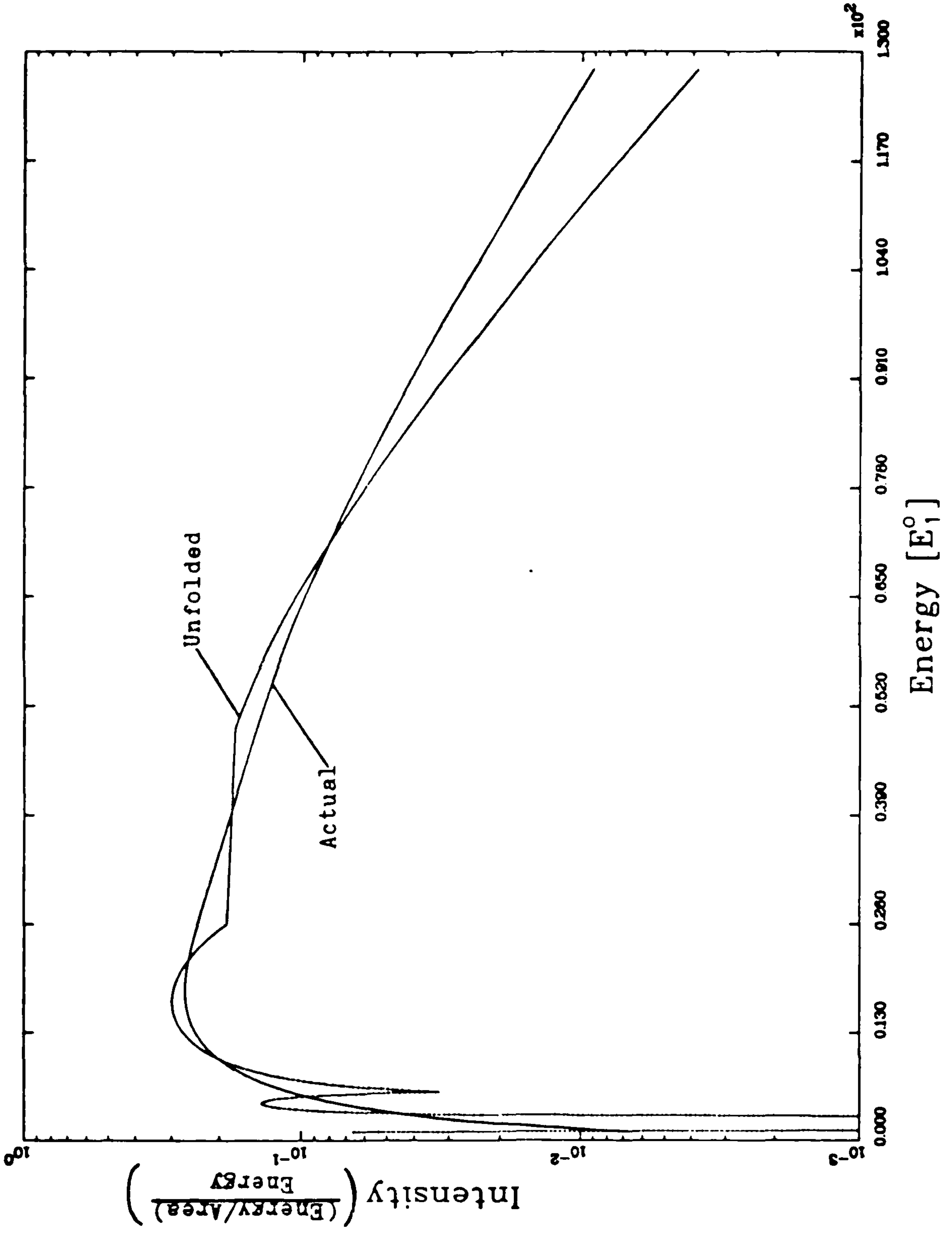


Figure 8. Semi-log plot of the Actual and Unfolded Spectra Versus Energy for Case PC3

TABLE VIII
Results of Fitting a Planckian Spectrum with Cubic Spline Basis Functions with Variable Knots (a,b,c)

Case		Parameters						χ^2
		BF1(d)		BF2		BF3		
Actual		4.0	5.0	3.0	10.0	7.0	15.0	
		Knots		Coefficients			T	
PC1	Initial	2.0	15.0	0.30	0.50	0.60	8.0	
		35.0		0.20				
	Unfolded	2.3	3.9	-0.28	-4.8	0.17	12.0	71.0
		30.0		0.22				
PC2	Initial	2.0	15.0	0.30	0.50	0.30	8.0	
		35.0	50.0	0.20	0.10			
	Unfolded	1.0	12.0	-0.43	-3.5		11.0	38.0
		34.0	49.0	0.12	-0.23	0.29		
PC3	Initial	2.0	10.0	0.30	0.40	0.50	8.0	
		15.0	35.0	0.30	0.20	0.10		
		50.0						
	Unfolded	1.8	2.2	0.097	-0.38	-0.22	12.0	16.0
		5.9	26.0	0.031	0.19	0.18		
		49.0						

a. $\sigma_i = 0.15$

b. Convergence criteria: $\chi^2 \leq 0.01$ or less than 0.1% change in χ^2 for successive iterations

c. Fixed knots at 0.0,1.0,128.0 are implicit in definition of these splines and are not listed

d. BF stands for basis function

This portion of the study demonstrates the strengths and weaknesses of the system very well. The results show how well the computer program can fit the tail of the planckian spectrum. This is due to the fact the fit between the last two knots in the cubic spline is defined as a planckian distribution. Secondly, the results demonstrate how the selection and quantity of knots effects the unfolded spectrum (i.e. the use of six or seven knots is not acceptable but the use of eight knots is acceptable.).

Finally, case PC3 depicts the artifacts that can be added to the unfolded spectrum as a result of the unfolding. This case is plotted in Figures 7 and 8. The final point to note is the negative coefficients. These negative coefficients are acceptable since the non-negativity constraint was imposed. This demonstrates the fact the computer treats the basis functions as merely mathematical functions and tries to add or subtract them as required.

The second "actual" spectrum studied was composed of cubic spline basis functions using seven knots. The spectrum was approximated using one, two, and three planckian basis functions for the unfolded spectrum. The results of the study are presented in Table IX and the best approximation to the actual spectrum is depicted by Figures 9 and 10. All plots of actual and unfolded spectra verses energy are presented in Appendix F.

The results of this study indicate that as the number of planckian basis functions is increased, the value for χ^2 is reduced. From the results, there appears to be an optimum number of knots that should be used to fit a given spectrum. However as seen from cases CP3 and CP4, the addition of another basis function may not improve the approximation of the actual spectrum. Also note the negative temperatures which have no physical meaning. However, the computer code is once again treating the basis functions strictly as mathematical functions and in this case the overall function may be negative. In order to require a positive temperature, a simple restraint on the temperatures in the computer code could be inserted, although this would result in a poorer agreement with the measurements.

Figure of Merit

In order to evaluate how well the actual spectrum was approximated by the unfolded spectrum, a study of eight figures of merit was conducted. This study is presented in Appendix G. The goal of the study was to determine a figure of merit that correlated with χ^2. Thus, an experimenter would have a reasonable idea of the goodeness of fit for the unfolded spectrum. However, the eight figures of merit studied do not appear to correlate well with χ^2. The main reason for the figures of merit not correlating with χ^2 was the calculation of the functions. χ^2

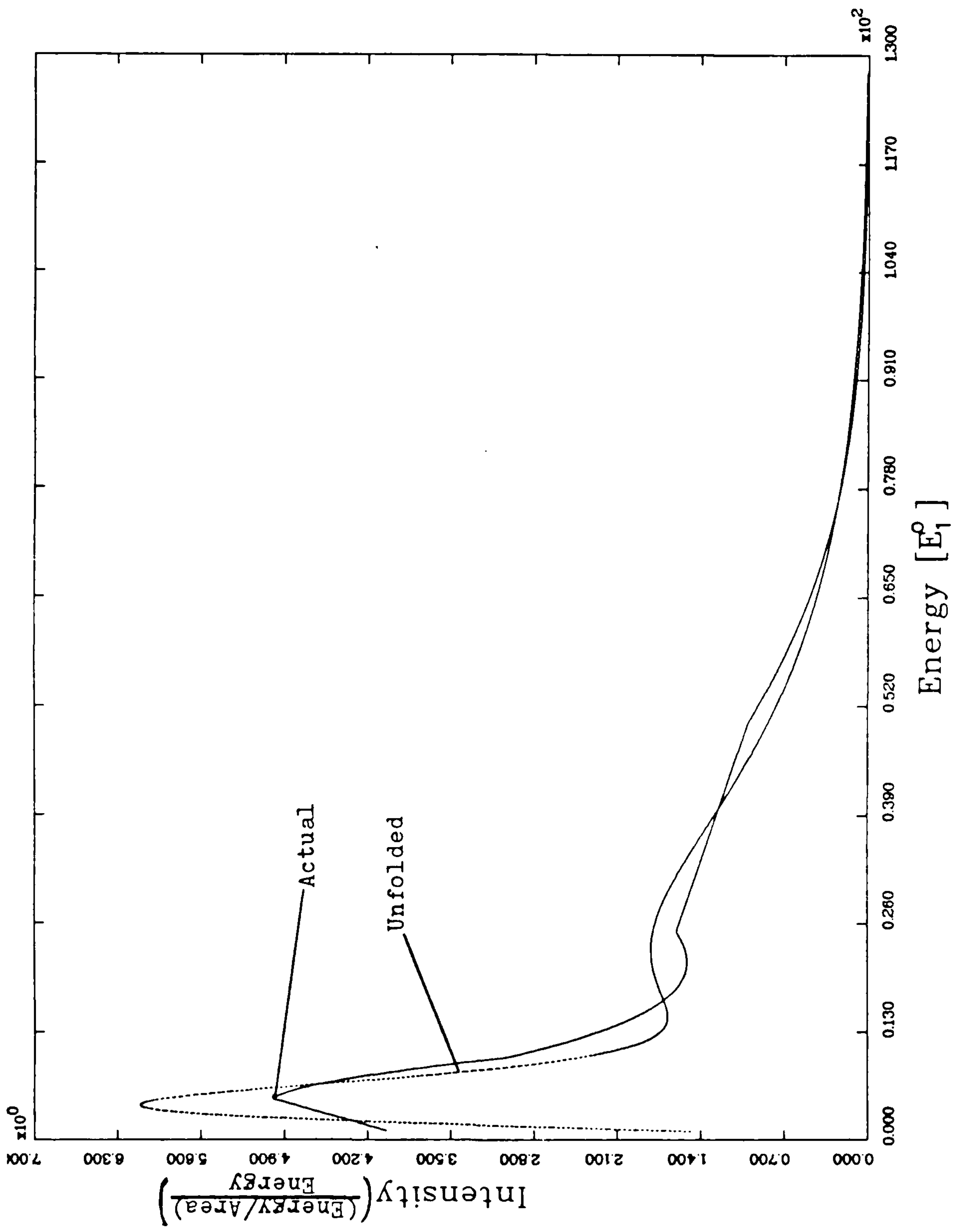


Figure 9. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case CP3

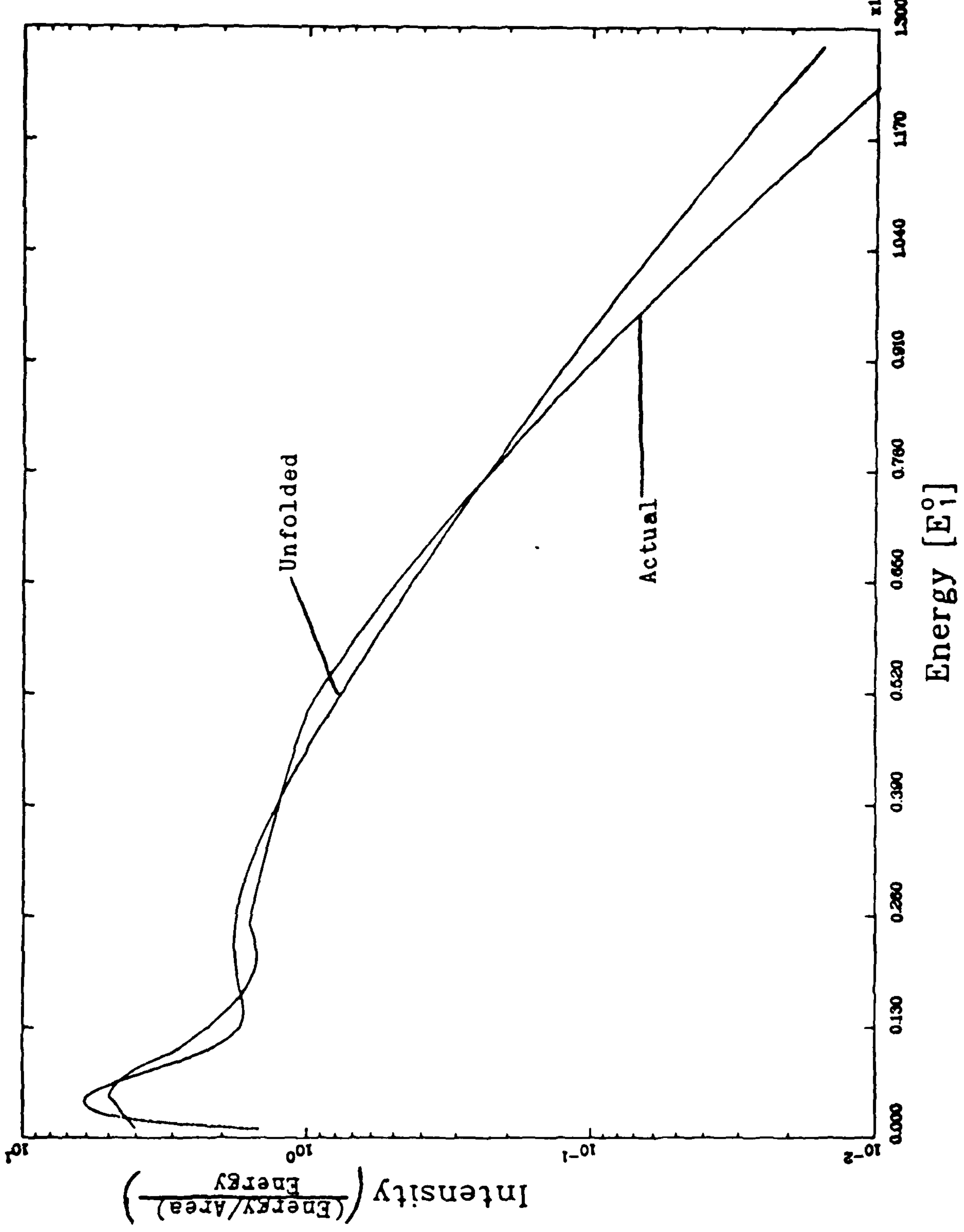


Figure 10. Semi-log Plot of the Actual and Unfolded Spectra Versus Energy for Case CP3

TABLE IX
Results of Fitting A Cubic Spline Spectrum With Planckian Basis Functions (a,b)

Case		Parameters								
		Knots				Coefficients				T
Actual(c)	5.0	10.0	25.0	50.0	4.0	5.0	3.0	1.6	1.0	10.0
		BF1(d)		BF2		BF3		Bf4		χ^2
		a	T	a	T	a	T	a	T	
CP1	Initial	3.0	2.0							
	Unfolded	67.0	2.5							380.0
CP2	Initial	3.0	2.0	4.0	6.0					
	Unfolded	43.0	1.6	76.0	9.9					20.0
CP3	Initial	3.0	2.0	4.0	6.0	2.0	8.0			
	Unfolded	39.0	1.5	40.0	7.0	39.0	12.0			17.0
CP4	Initial	3.0	2.0	4.0	6.0	2.0	8.0	1.0	3.0	
	Unfolded	36.0	-1.3	40.0	7.0	40.0	12.0	38.0	1.4	17.0

a. $\sigma_i = 0.15$

b. Convergence criteria: $\chi^2 \leq 0.01$ or less than 0.1% change in χ^2 for successive iterations

c. Fixed knots at 0.0,1.0,128.0 are implicit in definition of this spline and are not listed BF stands for basis function

was calculated based on the difference of the signals and thus was weighted more heavily at the low energies while the figures of merit were based on spectral values and received equal weight since the energy bins were equal. Thus, a good χ^2 would not correspond to a good figure of merit because a small difference in the ratios for the last detector would cause a large error in the figure of merits since the error would be spread over energies from $64E_i^0$ to $128E_i^0$.

VI. Summary and Conclusions

Summary

Validation Cases. The purpose of the validation cases was to ensure each element of the computer code was functioning properly and to note any flaws. The three validation cases presented do exactly that. The benchmark cases for the planckians and the cubic spline basis function with fixed knots demonstrate the power of this unfolding technique. These cases show how well an actual spectrum can be approximated if the type of basis function used to construct the unfolded spectrum is the best possible basis function and no measurement error, either real or simulated, is present. In other words, if an optimum basis function can be determined, the computer program is outstanding.

The cubic spline validation cases also demonstrate the dependence of the unfolded spectrum on the initial guess at the knot locations. Even with a poor choice for the initial knot locations, the method will attempt to unfold the actual spectrum. However, in general the variations in the knots will be so small as to require an unacceptable amount of computer time before the actual spectrum is unfolded. Thus, it is recommended that the predicted spectrum be used to determine the optimum knots for the initial guess at the unfolded spectrum. However, this does not guarantee the best unfold with the real data so the unfold may have to be repeated using different of knot locations

Test Cases. The test cases demonstrated that error in the measured signals (simulated measurement error in the b_i's in this study) prevent the "exact" actual spectrum from being unfolded. However, the general shape of the actual spectrum was unfolded in all cases. The test cases also demonstrated the artifacts introduced in the unfolded spectrum when attempting to approximate an actual spectrum that is not composed of the same set of basis functions as the unfolded spectrum. Thus, an experimenter can not unfold the actual spectrum but only an approximation to the actual spectrum and this unfolded spectrum will vary depending on the error in the data and the type of basis functions used. However, the general shape of the actual spectrum is returned

Conclusions

The results obtained from the test cases and validation cases demonstrated the following conclusions.

1. The final convergence and the rate of convergence of the unfold using cubic splines is dependent on the knot locations.
2. The cubic spline basis functions or local basis functions yielded a lower χ^2 than the planckian or global basis functions did when used to unfold data with simulated measurement error.
3. The total fluence of the spectrum is more accurate than the spectrum.
4. The shape of the spectrum is definable but the exact spectrum can not be unfolded.
5. The temperatures in the multi-temperature planckians do not represent the actual temperatures of the planckians.
6. The type of basis functions used will have a direct affect on the details one can unfold and on the artifacts added to the unfold spectra.

In addition, the validation cases and test cases studied demonstrate the usefulness of this data analysis method. The data analysis method be used in the following ways.

1. During the experimental planning phase, the computer program could be used to determine the best detector locations and response functions in order to achieve a reasonable unfold; and to optimize the number of detectors or channels in order to confirm the predicted spectrum or to determine variations from the predicted spectrum.

2. After the ideal detector locations are selected and the response functions are determined, the computer program can be used to evaluate the affect of various basis functions on the unfold and the affect of measurement error on the unfold.
3. The computer program can then be used to unfold the real data.
4. Finally, an analysis of the propagation of error can be conducted using the computer program in order to get uncertainty bounds for the unfolded spectrum and to determine the limitations of the unfold.

In order to convert Decon7.f to handle actual data as discussed above, four read statements would have to be inserted. These read statement would read the response functions, the predicted signals, the measured-to-predicted ratios, and the standard deviations of the measured-to predicted ratios. Thus, the computer code is capable of simulating test procedures or with three simple modifications, processing actual data.

VII. Recommendations

Recommendations

The following recommendations are made for continued study in this area. First, the simulated measurement error study should be continued in order to determine if the simulated measurement error has a more dramatic effect on any particular detector or detectors. In addition, this study could consider the possibility of one or more detectors failing and see how this failure effects the unfolded spectrum.

Secondly, the study should be continued to see how well the planckian and cubic spline basis functions can approximate other spectra. This study could also be used to determine if the artifacts produced correspond to the k-edges or energy ranges of certain detectors.

The third area of study could be a more detailed study of the relation of χ^2 to a given figure of merit. This study could be conducted in conjunction with a study to determine the error or uncertainty in the unfolded spectrum. This error analysis should include the methodology used in the deconvolution process as well as the error established in measuring the actual spectrum.

The fourth area that could be studied is the effect on the unfolded spectrum of varying the number of detectors used and/or their locations. These effects could be studied using various basis functions. Formulation, implementation, and evaluation of new and varied types of basis functions is a fifth area for future work.

Finally, a study could be conducted to determine how well the deconvolution process can resolve specific details in the actual spectrum. By selecting an actual spectrum and then adding a known function and varying its amplitude and width, a test procedure could be constructed to evaluate the resolving power of the deconvolution method.

Appendix A: Derivation of the Response Functions

Before the methodology used to develop the program could be implemented, typical response functions for the detectors had to be derived. In detecting pulsed x-ray radiation, a common technique uses a filter-fluorescer detection system. In principle, this system uses the k-edges of the filter and fluorescer to determine the inband response of the detector. This type of detection system was selected for the closed response function detectors. The open response function detectors used the same system without the filter.

The equation used to derive these response functions is (15)(4·244):

$$J_k(E,\theta)\,d\Omega\,dE = J(E)\tau_k(E)\omega_k\left(\frac{d\Omega}{4pi}\right)\left(\frac{dE}{E}\right)\left(\frac{S}{\cos\phi}\right) \tag{15}$$

$$\left(\sum_i W_i E_i\left(\frac{1}{\mu_i(E)}\right)\right)\left[1-\exp(-\mu_t(E)d)\right]$$

$$\mu_t(E) = \frac{\mu(E)}{\cos\phi}+\frac{\mu_i}{\cos(\pi-\theta)}$$

where

$J_k(E,\theta)$ = the intensity of the K series excited by radiation having an energy ranging from E to $E+dE$

$d\Omega$ = a solid angle

$\tau_k(E)$ = the coefficient of photoabsorption on the k shell

ω_k = the fluorescent yield

W_i = the probability ofthe appearance of the fluorescence quantum having an energy E_i

d = the foil thickness

$\mu(E)$ = reduction coefficient of the exciting radiation (considered mass attenuation coefficient)

$\mu_i(E)$ = reduction coefficient of the ith fluorescence line (also considered mass attenuation coefficient)

For the purpose of this study, Equation (15) was simplified as follows. First it was assumed that the experiment was arranged to consider only one angle and that this minimized $\mu_i(\cos(\pi - \theta))$. Also note that in the vicinity of the k-edge, $\mu(E)$ is much greater than μ_i. Thus, $\mu_i(E)$ can be approximated by $\frac{\mu(E)}{\cos\phi)}$. The second simplification was that only one fluorescence energy was studied. Thus, Equation (15) can be simplified to:

$$J_k(E) = (const)(\tau_k(E)) \left(\frac{dE}{E} \right) \left(\frac{1}{\left[\frac{\mu(E)}{\cos\phi} \right]} \right) (1 - \exp(\mu_i(E)d)) \tag{16}$$

Next, τ is proportional to $\frac{1}{(h\nu^0)^3}$ or $\frac{1}{E^3}$ (1:Section 1.8, page 2) and near the k-edge of a material τ is proportional to μ. This can be used to show $\mu_i(E)$ is proportional to $\left(\frac{E}{E^0} \right)$ where E^0 is the energy of the k-edge. Thus Equation (16) becomes:

$$J_k(E)dE = const \left(\frac{1}{E} \right) \left\{ 1 - \exp \left[-g \left(\frac{E^0}{E} \right)^3 \right] \right\} dE \tag{17}$$

where

g = some constant

The open response function of the detector can then be given by Equation (18) (4:245).

$$R^o(E) = \epsilon(d) J_k(E) \tag{18}$$

where

$\epsilon(d)$ = the efficiency of the detector. (assumed constant with energy)

Thus

$$R^{o}(E) = \left(\frac{1}{E}\right)\left\{1-\exp\left[-g\left(\frac{E^{0}}{E}\right)^{\beta}\right]\right\} \qquad (19)$$

The constant and $\epsilon(d)$ have been dropped due to the normalization of the predicted signal and the subsequent modification of the response function. Also note that the energy must be greater than the k-edge of the fluorescer in order for the fluorescence to occur.

The addition of a filter to this open response function detector only serves to attenuate the arriving radiation. Thus Equation (19) becomes:

$$R^{c}(E)=\begin{cases}\left(\frac{1}{E}\right)\left\{1-\exp\left[-n\left(\frac{E^{0}}{E}\right)^{\beta}\right]\right\}\left\{\exp\left[-m\left(\frac{E^{0}}{E}\right)^{\beta}\right]\right\} & E^{0}\leq E<E^{1}\\ \left(\frac{1}{E}\right)\left\{1-\exp\left[-n\left(\frac{E^{0}}{E}\right)^{\beta}\right]\right\}\left\{\exp\left[-mr\left(\frac{E^{0}}{E}\right)^{\beta}\right]\right\} & E\geq E^{1}\end{cases} \qquad (20)$$

where

n = constant
m = constant
r = ratio of mass attenuation coefficient values at k—edge
E^{1} = k—edge of filter

By conducting a study of mass attenuation coefficients, r was determined to range from 4 to 12. For this study a value of 6 was selected. Secondly, a resolving power of about 1.5 could be achieved if the E^{1} is twice the value of E^{0} for the detection system. This also established the limits for the inband response functions.

Finally, a parameter study was conducted to ensure at least 70% of the signal would be inband and that this signal would still be strong enough to be detectable over the other noise, such as scattered radiation, in the detection system. Based on the parameter study, the value of the constants was selected as follows: $g = 3.0$; $n = 2.0$. With a value of 2.0 for n, and $E^{1} = 2E^{0}$, $m = 0.25$

1/8 of n. With these values the open and closed inband response is 80% and 78% respectively and the total response is acceptable.

Appendix B: Derivation of the Cubic Spline Basis Functions

The cubic splines used in this study were four point Lagrangian interpolating splines and not the common B-spline. These cubic spline basis functions were a combination of four cubic splines. Also, a linear functions or a planckian function were required in certain segments of the spectrum. The lagrangian interpolating splines were a function of the knot locations chosen to construct the splines and the spectral intensity at these knots. In this study, the intensity was referred to as the expansion coefficient for the spline formed using that knot. The study required three fixed knots.The first knot is equal to 0.0; the second knot is equal to E_1^0 and the last knot is equal to $128.0{*}E_1^0$. Also, a minimum of six knots is required before a full cubic section is defined by the basis functions.

The number of basis functions is equal to the total number of knots minus one. Each knot location except the first and last knot location is used to define a specific basis function. Additionally, the next to last knot's spectral value is used to define two basis functions. This appendix will present the methodology used to develop the various basis functions.

The primary function used to define the basis function was the cubic spline function given by Equation (21)(2:85).

$$L_{4k}(E) = \prod_{\substack{i=0 \\ i \neq k}}^{4} \frac{(E - knot_i)}{(knot_k - knot_i)} \tag{21}$$

where

E = energy
$knot_k$ = primary knot
$knot_i$ = other knots in the function

In this study, four functions, $L_{4k}(E)$'s, were used to define the basis function for the primary knot locations. This basis function covered a range from the energy of two knots to the left of the primary knot location up to two knots to the right of the primary knot. In other words if the given knot is knot(x), the energy range covered from knot(x-2) to knot(x+2). Figure 11 depicts the formation of a cubic spline basis function using the four $L_{4,k}(E)$'s described by Equation (21) and expanded in Equations (22) to (25).

$$f(1) = \left\{ \frac{(x-x1)(x-x2)(x-x3)}{(x4-x1)(x4-x2)(x4-x3)} \right\} \qquad x2 \leq x < x3$$

$$f(2) = \left\{ \frac{(x-x2)(x-x3)(x-x5)}{(x4-x2)(x4-x2)(x4-x5)} \right\} \qquad x3 \leq x < x4$$

$$f(3) = \left\{ \frac{(x-x31)(x-x5)(x-x6)}{(x4-x3)(x4-x5)(x4-x6)} \right\} \qquad x4 \leq x < x5$$

$$f(4) = \left\{ \frac{(x-x5)(x-x6)(x-x7)}{(x4-x5)(x4-x6)(x4-x7)} \right\} \qquad x5 \leq x < x6$$

The solid line portions of each of the four $L_{4,k}$ functions depicted in Figure 11(a to d) are added to form the final cubic basis function depicted in Figure 11e. A *cubic spline basis function is formed in the same manner for each knot* location unless the function is modified as discussed below.

Three modifications were made to these cubic functions. First, the detectors are unable to detect energy below E_1^0 or above $128{*}E_1^0$. Thus, these values were not computed in the basis functions used to define the unfolded spectrum. Secondly, four complete cubics can not be evaluated between the next to last knot and the last knot. Thus, a planckian fit, as given by Equation (6), was used in this section. Also, four complete cubics could not be calculated from knot two to knot three or after the planckian modification, from two knots from the end to one knot before the end. In order to allow all other sections to be fit as cubics and to correct these two sections, a linear fit was used.

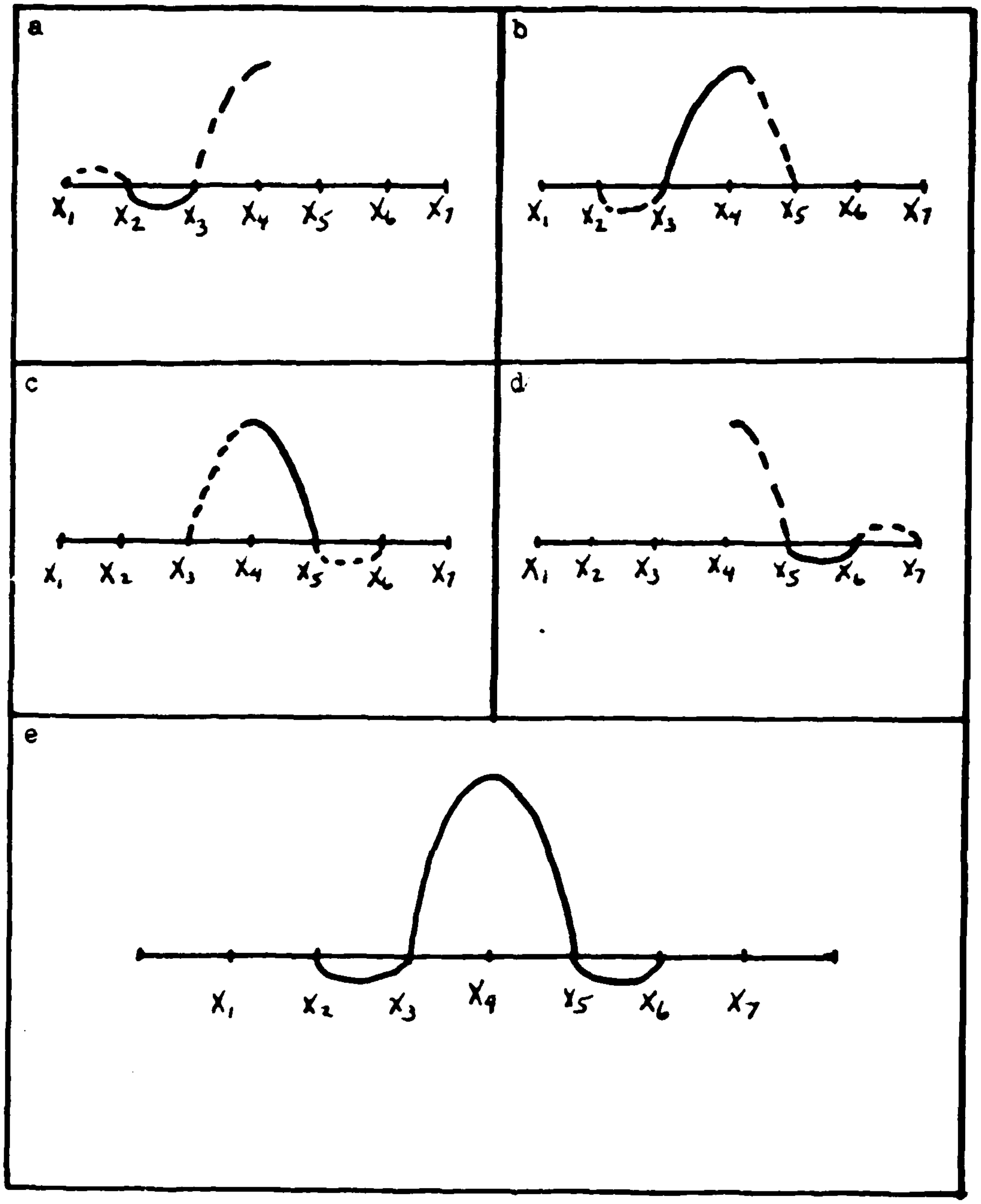


Figure 11. Development of the Cubic Spline Basis Function

In order to visualize these modifications, Figure 12 depicts the five basis functions used for the case consisting of six knots. The basis functions cover the following knot locations: basis function 1, 1.0 to 50.0; basis function 2, 1.0 to 50.0; basis function 3, 10.0 to 80.0; basis function4, 10.0 to 80.0; and basis function 5, 80.0 to 128.0. Note the error in the functions at knot5. This is due to the step size used to produce the figure. By using a smaller step size to calculate the basis functions, a smooth set of basis functions can be produced at this last knot.

Table X presents the basis functions used and the knots and type of functions used to define the basis functions.

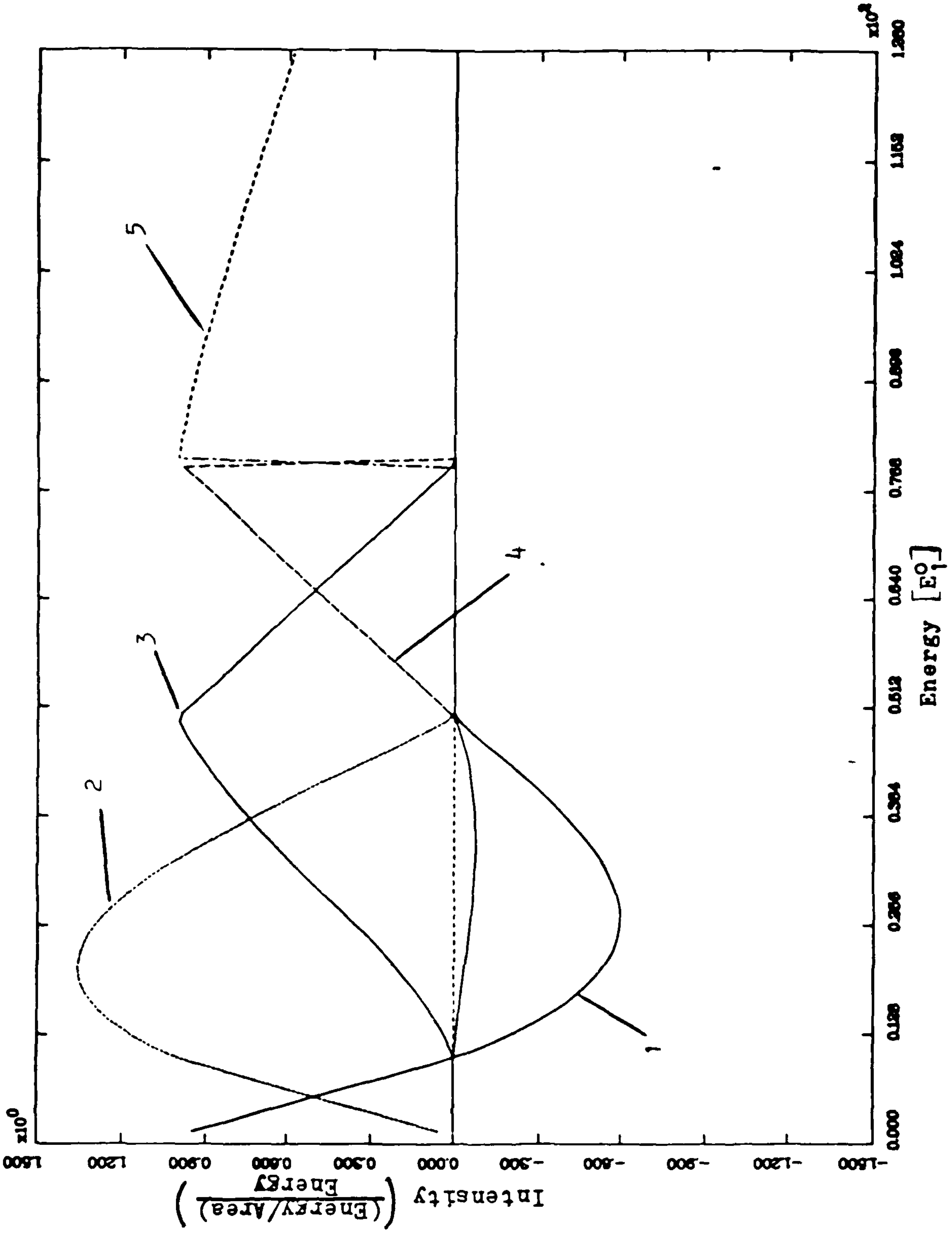


Figure 12 Cubic Basis Functions for the Case Using 6 Knots

TABLE X
Knots and Functions Used to Define the Cubic Spline Basis Functions(a,b)

Basis Function	Primary Knot(c)	Energy Range(d)	Knots Used	Type Function
1	2	2-3	2,3	linear
		3-4	2,3,4,5	cubic
2	3	2-3	2,3	linear
		3-4	2,3,4,5	cubic
		4-5	3,4,5,6	cubic
3	4	3-4	2,3,4,5	cubic
		4-5	3,4,5,6	cubic
		5-6	4,5,6,7	cubic
4(e)	5	3-4	2,3,4,5	cubic
		4-5	3,4,5,6	cubic
		5-6	4,5,6,7	cubic
		6-7	5,6,7,8	cubic
N-3	N-2	(N-2)-(N-1)	N-3,N-2,N-1,N	cubic
		(N-3)-(N-2)	N-4,N-3,N-2,N-1	cubic
		(N-4)-(N-3)	N-5,N-4,N-3,N-2	cubic
N-2	N-1	(N-1)-N	N-1,N	linear
		(N-2)-(N-1)	N-3,N-2,N-1,N	cubic
		(N-3)-(N-2)	N-4,N-3,N-2,N-1	cubic
N-1	N	(N-1)-N	N-1,N	linear
		(N-2)-(N-1)	N-3,N-2,N-1,N	cubic
N	N	N-N+1	N-1	planckian

a. N equals the number knots minus 1

b. If less than eight knots are used, the basis functions must be modified to ensure only a linear fit is used between knots 2 and 3 and knots N and N-1.

c. Primary knot value is equal to one in all cases. All other knots are equal to zero. Basis function is then expanded using the expansion coefficient for the primary knot.

d. Energy Range is considered as energies from knot to knot .

e. Also used for basis functions 5 to N-4 with primary knot equal to basis function plus one.

Appendix C: Computer Program Documentation

Prose

Decon7.F is a fortran 77 computer program that is written off the UNIX ELXSI BSD (4.2) operating system. The computer program is currently available on the ICC computer system at the Air Force Institute of Technology. Decon7.f is a computer program that can be used to approximate the actual radiation spectrum in the Fredholm integral equation. Decon7.f can also test various types of basis functions for defining the unfolded spectrum. With minor modification, the response functions or number of detectors utilized in the unfolded process can also be changed. This allows the program to be used to study the effect of the detectors and the type of response function on the unfolding process. The methodology used to develop this program is discussed in detail in Section III.

The program is based on three main assumptions. First, the energy range considered is from E_1^0 to $128E_1^0$. Secondly if the actual and predicted are defined by the computer program vice being input, the program assumes the calibrated and actual response functions are equal. Finally, decon7.f assumes 20 detectors are utilized to measure the spectra.

Currently, the program is limited to planckian and cubic spline basis functions for defining the unfolded spectrum. However, a new basis function subroutine can easily be inserted for defining the unfolded spectrum. Also, since the function is non-linear, the possibility exists for the computer program to begin to loop between certain values of χ^2. This could be corrected by either adding a maximum number of iterations allowed or by stopping the computer program and then restarting the program with a higher convergence criteria for χ^2.

Decon7.f is run by compiling and then executing the source code. Decon7 will then prompt the user for all input required and list all options available.

Pseudo Code for Main Program

```
Input $e_1^0$.
Input energy bin width for calculating integrals.
Define $E_i^0$ and $E_i^1$ for all detectors with respect to $E_1^0$.
Calculate response functions using Equations (3) and (4).
Input type of predicted spectrum to be used. (planckian, cubic or other.)
If planckian, then
    Input number of basis functions
    Input coefficient and temp. for each basis function
    Calculate predicted spectrum by calling bbfunc
End if
If cubic then
    Input the number of knots
    Input knot locations
    Input expansion coefficients
    Input the planckian temperature
    Calculate predicted spectrum by calling csfunc
End if
If other then
    Read predicted spectrum from input file
End if
Calculate $Y_i^p$ using Equation (8)
Divide $R_i(E)$ by $Y_i^p$ to normalize $Y_i^p$
```

```
Define the actual spectrum (use the same method as defining the predicted method).
Calculate $b_i$ using Equation (10)
Input $\sigma_i$ for all detectors
Option to add normal noise distribution to $b_i$.     (Input 1 for yes)
If yes then
    Input seed number for random number generator
    Calculate $Z_i$ using Equation (14).
    Add $Z_i$ to $b_i$
End if
Option for flux non-negativity constraint (Input 1 for yes)
Input type of basis function to define the unfolded spectrum (planckian or cubic)
If planckian then
    Input the number of basis functions
    Input the coefficient and temperature for each basis function
End if
If cubic then
    Input the number of knots
    Input the knot locations
    Input the expansion coefficients
    Input the planckian temperature
End if
Initialize the H matrix for the Fletcher-Powell method to the identity matrix.
```

```
Input the initial guess at the functional lower bound
Calculate the unfolded spectrum by calling bbfunc or csfunc
Calculate $c_i$ using Equation (11)
Calculate $\chi^2$ using Equation (12)
While $\chi^2 >$ convergence criteria
    Call minimize
    Recalculate $\chi^2$
    Print iteration #, $\chi^2$, and parameters  to screen
End while
If $\chi^2 \leq$ convergence criteria
    Write actual and unfolded spectrum into an output file
End if
End of main program
```

Pseudo Code for Subroutine bbfunc

```
Calculate spectrum using equation (6) and apply flux non-negativity options
if applicable
Return spectrum values to calling module.
```

Pseudo Code for Subroutine csfunc

```
Calculate spectrum using cubic splines in Appendix B and apply flux non-
negativity option if applicable.
Return spectrum values to calling module.  flux
```

Pseudo Code for Subroutine minimize (5:75-76)

```
For p=1 to np
    Call gradient $c_i$
    Calculate gradient of $\chi^2$
Next p
Multiply gradient matrix by H matrix to find search direction
Normalize search direction to find unit search direction
Call subroutine linesearch
Reset H matrix to identity matrix if slope of $\chi^2 > 0.0$ and recalculate search direction.
Calculate new parameters for unfolded spectra
Calculate new H matrix
Call bbfunc or csfunc to calculate new spectrum based on new parameters
Calculate $pc_i$ for new parameters using equation (11)
Let $pc_i = c_i$
Return new parameter and $c_i$ to main program,
```

Pseudo Code for Subroutine linesearch (5:77-80)

```
Calculate functional value (ho) and slope (mo) with given parameters
If slope > 0.0
    Return and reset H matrix
End if
Calculate t2
Calculate functional value (h2) and slope (m2) using parameter + t2 * unit search direction.
```

```
If $m2 > 0.0$
    Use the two slopes and functional values previously calculated to fit a
    cubic and determine the line search distance
End if
If $m2 < 0.0$ and $h2 > h0$
    Cut the search interval in half and recalculate m2, h2
End if
If $m2 = 0.0$ and $h2 < h0$
    Line search distance equals t2.
End if
If $m2 < 0.0$ and $h2 < h0$
    Double search distance and let $m0 = m2$, $h0 = h2$, $t0 = t2$
    Recalculate m2 and h2
End if
If $m2 = 0.0$ and $h0 = h2$ (modification to reference)
    Line search distance equals to
End if
Return line search distance to minimize subroutine
```

Pseudo Code for Subroutine calcfunc

```
Calculate new parameters
        (tparameter = parameter + t* unit search direction)
Call bbfunc or csfunc to determine spectrum values
Calculate $c_i$ based on new parameters using equation (11)
Calculate $\chi^2$ using equation (12).
```

```
For p = 1,np
    Call gradient $c_i$
    Calculate gradient of $\chi^2$
Next p
For p = 1,np
    slope = slope + gradient $\chi^2$ * unit search direction
Next p
Return slope and function value, $\chi^2$, to linesearch subroutine
```

Pseudo code for Subroutine gradientc

```
Let parameter = parameter + delta* parameter with respect to the given parameter
Call bbfunc or csfunc to determine new spectrum values
Calculate new $c'$, let it equal $f_i$ using equation (11)
Calculate gradient $c_i$ using $c_i$ and $f_i$
Return the value for the gradient of c
```

Source Code for Decon7.f

See Appendix D

List of Variables used in the Source Code

See tab 1 to this appendix

Inputs

Inputs required for Decon 7.f are listed in Table XI.

TABLE XI

List of Input Variables

Var.	Definition
E_1^0	k-edge of fluorescor for 1st detector
y	energy bin width for calculating integrals
type*	type of basis function used to define predicted spectrum
dum1*	# of planckian basis functions; number of knots for cubic; or stop feature for reading input file
pparam(j)*	coefficient and temperatures for planckian basis function; or knots, expansion coefficients and temperature for cubic
dum1	toggle for simulated measurement error
dum2	seed for random number generator
non	toggle for non-negativity option
type	type of basis function used to define the unfolded spectrum
σ_i	standard deviation for b_i
nj	number of planckian basis functions
nkn	number of knots for cubic basis function
param(j)	coefficients and temperatures for each planckian basis function or knots, expansion coefficient and temperature for cubic basis function
l	initial guess at the functional lower bound.

* also required for the actual spectrum

Outputs

The output occurs in two forms. First, the values for χ^2 and all parameters are printed to the computer screen during each iteration as are the final parameters. Secondly, the actual spectrum and unfolded spectrum values at each energy location are printed into an output file labeled output.

File Structure

The program uses one output file and can use two input files. In order to run the program both input files must be defined. However, the two input files may be empty if they are not required to define the actual or predicted spectrum. The file structure for the output file is a two column table with energy and intensity at that specific energy. The first half of the file contains the actual spectrum. These energies begin with the maximum energy $128E_1^0 - \frac{Y}{2}$ and decrease in steps of the bin width (y) to a value of $E_1^0 + \frac{y}{2}$. The second half of the file consists of the unfolded spectrum and begins with energy $E_1^0 + \frac{y}{2}$ and increases in steps of y to the maximum energy. The file structure for both input files consist of a single column of energy intensities for energies from $E_1^0 + \frac{y}{2}$ to the maximum energy in steps of y, the bin width.

Validation Cases

See Sections V and VI.

Test Cases

See Sections V and VI.

Tab 1: List of Variables (a)

Variable in code	Symbol in Text	Definition	Modules(b)
a	--	dummy matrix used to calculate H	4
aknot1	--	knot 0.0 for actual spectrum	1
aknot1	--	knot Eo(1) for actual spectrum	1
aknot15	--	knot 128Eo(1) for actual spectrum	1
aparam(j)	--	jth parameter for actual spectrum	1
b(i)	b_i	measured-to-predicted ratio	1,4,5,6
c(i)	c_i	unfolded-to-predicted ratio	1,4
c(i)	--	c(i) for appropriate spectrum	7
chi2	χ^2	chi squared	1
coef(p)	--	expansion coefficient knot p-1	3
const	--	expansion coefficient for planckian	3
const	--	dummy variable for summing random numbers	1
d(i)	d_i	integral of Ri(E) Sp(E) dE	1
delta	--	step size to calculate gradient	7
dum	--	dummy variable	3
dum1	--	dummy variable	4
dum1	--	dummy variable	5
dum1,dum2	--	dummy variables	1
dum3	--	% change inχ^2	1
dummy	--	dummy variable	1
e	E	energy counter	1,2,3
eo(i)	E_i^0	k-edge of fluorescer for ith detector	1,2,3, 4,5,6,7
el(i)	E_i^1	k-edge of filter for ith detector	1
f(p)	--	c(i) for newpar(p)	7
g(p)	--	gradient of χ^2 for old parameters	4
gprime(p)	--	gradient χ^2 for new parameters	4
gradc	--	gradient of c(i) for appropriate spectrum	7
gradc(p)	--	gradient c(i) with respect to parameter p	4
gradc(p)	--	gradient of tc(p)	6
grchi2(p)	--	gradient of χ^2 for tparam	6
grchi2(p)	--	gradient χ^2 with respect to parameter p	4
h	--	correction matrix H	1,4,5
ho	--	functional value at (Param+to*s)	5
ht	--	dummy functional value variable	5,6
h2	--	functional value at (Param+t2*s)	5
i	--	counter	1,4,6,7
it	--	iteration counter	1
j	--	counter	1,2,3,4
k	--	counter	1,2,3, 4,6,7
kc	--	planckian conversion constant (temp. to energy)	1,2,3, 4,5,6,7
kn(p)	--	knot number p	3
kn(15)	--	dummy variable for knot location	4
knot1	--	knot 0.0, unfolded spectrum	1,3,4, 5,6,7
knot2	--	knot Eo(1), unfolded spectrum	1,3,4, 5,6,7

Variable in code	Symbol in Text	Definition	Modules(b)
knot15	--	knot 128Eo(1), unfolded spectrum	1,3,4, 5,6,7
l	--	functional lower bound	1,4,5
m	--	dummy slope variable	5,6
mo	--	slope at (param + to*s)	5
mu	--	variable for cubic fit	5
m2	--	slope at (param + t2*s)	5
newpar(p)	--	param(p) + (delta)*(param(p))	7
ni	--	number of detectors	1,4,5, 6,7
nj	--	number of planckian basis functions	1,2,4,
nk	--	number of energies	1,2,3, 4,5,6,7
nkn	--	number of knots unfolded spectrum	1,3,4, 5,6,7
np	--	number of parameters	1,4,5, 6,7
non	--	toggle for flux non-negativity option	1,2,3 4,5,6,7
norms	--	norm of sprime	4
num	--	dummy variable	3
o	--	counter	1,4
p	--	counter	1,4,6,7
param(j)	--	jth parameter for unfolded spectrum	1,2,3, 4,5,6,7
pc(i)	--	c(i) for new parameters	4
pi	π	pi	1,2,3, 4,5,6,7
pknot1	--	knot 0.0 for predicted spectrum	1
pknot2	--	knot Eo(1) for predicted spectrum	1
pknot15	--	knot 128Eo(1) for predicted spectrum	1
planck	--	planckian distribution	3
pparam(j)	--	jth parameters for predicated spectrum	1
pprime(p)	--	new parameter	4
q	--	dummy matrix used to calculate H	4
r(i,k)	$R_i(E)$	response function of ith detector kth energy	1,4,5, 6,7
rand	--	random number generator	1
reset	--	reset counter for H matrix	4,5
s(p)	--	unit normal search direction for parameter p	4,5,6
sa(k)	$S_a(E)$	actual spectrum kth energy	1
sig	--	dummy input for σ_i	1
sigma(i)	σ_i	standard deviation in b_i	1,4,5,6
slope	--	slope for linear fit	3
sp(k)	$S_p(E)$	predicted spectrum kth energy	1
sprime(p)	--	search direction for parameter p	4
srand	--	initializes random number generator	1
su(k)	$S_u(E)$	unfolded spectrum kth energy	1,2,3
su(k)	--	dummy variable unfolded spectrum	4
su(k)	--	unfolded spectrum for tparam(p)	6
su(k)	--	unfolded spectrum for newpar(p)	7

Variable in code	Symbol in Text	Definition	Modules(b)
sumi	--	dummy variable	4
sumj	--	dummy variable	4
sumk	--	dummy variable	4
sumo	--	dummy variable	4
sump	--	dummy variable	4
sumran	--	sum of random numbers	1
t	--	temperature of planckian	3
t	--	step size in search direction	4,5,6
tc(p)	--	c(i) for tparam(p)	6
to	--	step size used to determine ho,mo	5
tparam(p)	--	parameter(p) +t*s	6
type	--	type of basis function	1,4,5, 6,7
t1	--	calculated step size	5
t2	--	step size used to determine h2,m2	5
v	--	dummy matrix used to calculate H	4
w	--	variable for cubic fit	5
x	--	dummy variable	1,2,3
x	--	parameter transfer counter	7
y	--	energy bin width	1,2,3, 4,5,6,7
z	--	dummy matrix used to calculate H	4
z	--	variable for cubic fit	5

a. If a parameter is listed once for several modules, the parameter is passed between the modules. If the parameter is listed separately for each module, the parameter is only common to that module.

b. Module listing: 1, Main; 2, bbfunc; 3, csfunc; 4, minimize; 5, linsearch; 6, calcfunc; 7, gradientc

Appendix D: Computer Program Source Code

```
c     THIS PROGRAM IS THE FINAL VERSION OF DECON.   IT USES
c     THE FLETCHER-POWELL METHOD FOR MINIMIZING CHI2 WHILE
c     USING A CUBIC LINE SEARCH ROUTINE.  ALL INPUTS ARE
c     REQUIRED TO BE INPUT FROM THE KEYBOARD.  THE CUBIC
c     BASIS FUNCTIONS INCLUDE THE KNOTS AS PARAMETERS.
c     THIS PROGRAM USES NUMERICAL DECONVOLUTION TO
c     APPROXIMATE AN ACTUAL RADIATION SPECTRUM WHEN GIVEN THE
c     MEASURED SIGNAL, A PREDICTED SPECTRUM, AND RESPONSE
c     FUNCTION OF THE DETECTOR(S).  THE PROGRAM ASSUMES
c     13 CLOSED RESPONSE DETECTORS AND 7 OPEN RESPONSE
c     DETECTORS.  THE PROGRAM USES AN ENERGY RANGE OF Eo(1)
c     TO 128*Eo(1)
      real*8 pparam(30),pknotl,pknot2, pknot15,aparam(30)
      real*8 aknotl,aknot2,aknot15
      real*8 chi2,eo(20),el(20),r(20,1280),e,pi,kc
      real*8 sa(1280),sp(1280),d(20),b(20),sigma(20)
      real*8 sumran,sig,dum3,dummy,x,su(1280)
      real*8 l,param(30),c(20),y,h(30,30)
      real*8 knotl,knot2,knot15,srand,rand,const
      integer nkn,type,it,non,i,k,j,ni,nk,nj,duml,dum2,o,p,np
      common eo(20)
      open (unit=7,file='output',status='new')
      open (unit=2,file='inputl',status='old')
      open (unit=3,file='input2',status='old')
      rewind (2)
      rewind (3)
      print *,'INPUT EO(1)'
      read *,eo(1)
c     NI=THE NUMBER OF INSTRUMENTS OR DETECTORS
c     NK=THE NUMBER OF ENERGY BINS USED FOR INTEGRATION
c     Y=THE WIDTH OF THE ENERGY BIN
      ni=20
      print *, 'INPUT THE DESIRED ENERGY BIN WIDTH, MINIMUM'
      print *, 'BIN WIDTH IS 0.1'
      read *,y
      nk=int((128.0*eo(1)-eo(1))/y)
      pi=3.1415
      kc=1.0
c     DEFINE ALL EO'S WITH RESPECT TO EO(1)
      do 5 i=2,7
        eo(i)=2.0*eo(i-1)
 5    continue
      eo(8)=1.5*eo(1)
      do 10 i=9,13
        eo(i)=2*eo(i-1)
 10   continue
      do 15 i=14,20
        eo(i)=eo(i-13)
 15   continue
```

```
c     DEFINE E1(I)WITH RESPECT TO EO(I)
      do 20 i=1,ni
        e1(i)=2.0*eo(i)
 20   continue
c     DEFINE ALL RESPONSE FUNCTIONS AT SPECIFIED ENERGIES
      do 30 i=1,ni
        e=eo(1)-y/2.0
        do 25 k=1,nk
          e=e+y
          if (e.gt.eo(i)) then
            if (i.lt.13) then
              if (e.lt.el(i)) then
                r(i,k)=(1.0/e)*(1.0-exp(-2.0*(eo(i)/e)**3))
                r(i,k)=r(i,k)*exp(-0.25*(el(i)/e)**3)
              else
                r(i,k)=(1.0/e)*(1.0-exp(-2.0*(eo(i)/e)**3))
                r(i,k)=r(i,k)*exp(-1.5*(el(i)/e)**3)
              end if
            else
              r(i,k)=(1.0/e)*(1.0-exp(-3.0*(eo(i)/e)**3))
            end if
          else
            r(i,k)=0.0
          end if
 25     continue
 30   continue
c     DEFINE THE PREDICTED SPECTRUM
      print *,'INPUT THE TYPE OF FUNCTION TO BE USED TO MODEL
      print *,'THE PREDICTED SPECTRUM'
      print *,'     1=PLANCKIAN BLACK BODIES'
      print *,'     2=CUBIC SPLINES'
      print *,'     3=OTHER'
      read *,type
      if (type.eq.1) then
        print *,'INPUT THE NUMBER OF PLANCKIAN BASIS'
        print *,'FUNCTIONS TO USE'
        print *,'(MUST BE AN INTEGER LESS THAN 15)'
        read *,duml
        print *,'INPUT THE COEF. AND TEMP. FOR EACH BASIS'
        print *,'FUNCTION SEPERATED BY A COMMA (TEMP IS IN'
        print *, 'UNITS OF EO(1))'
        do 33 j=1,duml
          print *,'BASIS FUNCTION NUMBER',j
          read *,pparam(j),pparam(j+duml)
 33     continue
        non=1.0
        call bbfunc(pparam,sp,y,nk,duml,pi,kc,non)
      end if
      if (type.eq.2) then
        print *,'INPUT THE NUMBER OF KNOTS TO BE USED TO FIT'
        print *,'THE CUBIC SPLINES (MUST BE AN INTEGER'
        print *,'6<=#KNOTS<=15) NOTE:0.0'
        print *,'IS COMPUTED AS THE FIRST KNOT, EO(1) IS'
        print *,'COMPUTED AS KNOT 2, AND 128.0*EO(1) IS'
        print *,'COMPUTED AS THE LAST KNOT.  INCLUDE THESE'
        print *,'KNOTS WHEN COUNTING YOUR KNOTS,AND ENSURE'
        print *,'THE KNOTS ARE INPUT FROM THE MINIMUM TO THE'
        print *,' MAXIMUM.'
        read *,dum1
        print *,'INPUT THE KNOTS'
        pknot1=0.0
        print *,'KNOT 1=',pknot1
        pknot2=eo(1)
        print *,'KNOT 2=',pknot2
        do 37 j=1,dum1-3
          print *,'input knot ',j+2
          read *,pparam(j)
37      continue
        pknot15=128.0*eo(1)
        print *,'KNOT',dum1,'=',pknot15
        do 38 j=1,dum1-2
          if (j.ne.1) then
            print *,'INPUT THE EXPANSION COEF. FOR'
            print *,' KNOT=',pparam(j-1)
          else
            print *,'INPUT THE EXPANSION COEF. FOR'
            print *,' KNOT=',pknot2
          end if
          read *,pparam(dum1-3+j)
38      continue
        print *,'INPUT THE TEMP. TO BE USED TO CONSTRUCT A'
        print *,'PLANCKIAN FIT BETWEEN THE LAST 2 KNOTS,'
        print *,' TEMP IS IN UNITS OF Eo'
        read *,pparam(2*dum1-4)
        non=1.0
        call csfunc(pparam,sp,dum1,nk,y,non,pknot1,
     *  pknot2,pknot15,eo,pi,kc)
      end if
      if (type.eq.3) then
        print *,'Sp(E) WILL BE READ FROM FILE input1, THE'
        print *,'FILE SHOULD CONSIST OF SPECTRUM VALUES FROM'
        print *,'(EO(1)+BIN WIDTH/2.0)'
        print *,'TO 128.0*EO(1) IN STEPS OF THE BIN WIDTH'
        print *,'INPUT A 1 TO CONTINUE OR A 2 TO STOP'
        read *,dum1
        if (dum1.eq.2) then
          go to 151
        end if
        e=eo(1)-y/2.0
        do 44 k=1,nk
          e=e+y
          read (2,*) sp(k)
44      continue
      end if
```

```
c     CALCULATE THE NORMALIZATION CONSTANT FOR D(I) AND
c     RENORMALIZE THE RESPONSE FUNCTIONS
      do 55 i=1,ni
        do 45 k=1,nk
          d(i)=d(i)+sp(k)*r(i,k)*y
 45     continue
        do 50 k=1,nk
          r(i,k)=r(i,k)/d(i)
 50     continue
 55   continue
c     DEFINE THE ACTUAL SPECTRUM
      print *,'INPUT THE TYPE OF FUNCTION TO BE USED TO'
      print *,'MODEL THE ACTUAL SPECTRUM'
      print *,'      1=PLANCKIAN BLACK BODIES'
      print *,'      2=CUBIC SPLINES'
      print *,'      3=OTHER'
      read *,type
      if (type.eq.1) then
        print *,'INPUT THE NUMBER OF PLANCKIAN BASIS'
        print *,'FUNCTIONS TO USE'
        print *,'(MUST BE AN INTEGER LESS THAN 15)'
        read *,duml
        print *,'INPUT THE COEF. AND TEMP. FOR EACH BASIS'
        print *,'FUNCTION SEPERATED BY A COMMA (TEMP IS IN'
        print *,'UNITS OF EO(1))'
        do 59 j=1,duml
          print *,'BASIS FUNCTION NUMBER',j
          read *,aparam(j),aparam(j+duml)
 59     continue
        non=1.0
        call bbfunc(aparam,sa,y,nk,duml,pi,kc,non)
      end if
      if (type.eq.2) then
        print *,'INPUT THE NUMBER OF KNOTS TO BE USED TO FIT'
        print *,'THE CUBIC SPLINES (MUST BE AN INTEGER'
        print *,'6<=#KNOTS<=15) NOTE:0.0'
        print *,'IS COMPUTED AS THE FIRST KNOT, EO(1) IS'
        print *,'COMPUTED AS KNOT 2, AND 128.0*EO(1) IS'
        print *,'COMPUTED AS THE LAST KNOT.'
        print *,'INCLUDE THESE KNOTS IN THE COUNT OF YOUR'
        print *,'KNOTS AND ENSURE THE KNOTS ARE INPUT FROM'
        print *,'THE MINIMUM TO THE MAXIMUM'
        read *,duml
        print *,'INPUT THE KNOTS'
        print *,'KNOT 1=0.0'
        aknot1=0.0
        aknot2=eo(1)
        print *,'KNOT 2=',aknot2
        do 62 j=1,duml-3
          print *,'input knot ',j+2
          read *,aparam(j)
 62     continue
```

```
        aknot15=128.0*eo(1)
        print *,'KNOT',duml+3,'=',aknot15
        do 63 j=1,duml-2
          if (j.ne.1) then
            print *,'INPUT THE EXPANSION COEF. FOR'
            print *,'KNOT=',aparam(j-1)
          else
            print *,'INPUT THE EXPANSION COEF. FOR'
            print *,'KNOT=',aknot2
          end if
          read *,aparam(duml-3+j)
 63     continue
        print *,'INPUT THE TEMP. TO BE USED TO CONSTRUCT A'
        print *,'PLANCKIAN FIT BETWEEN THE LAST 2 KNOTS,'
        print *,'TEMP IS IN UNITS OF Eo'
        read *,aparam(duml*2-4)
        non=1.0
        call csfunc(aparam,sa,duml,nk,y,non,
     *  aknot1,aknot2,aknot15,eo,pi,kc)
      end if
      if (type.eq.3) then
        print *,'Sa(E) WILL BE READ FROM FILE input2, THE'
        print *,'FILE SHOULD CONSIST OF SPECTRUM VALUES FROM'
        print *,'(EO(1)+BIN WIDTH/2.0)'
        print *,'TO 128.0*EO(1) IN STEPS OF THE BIN WIDTH'
        print *,'INPUT A 1 TO CONTINUE OR A 2 TO STOP'
        read *,duml
        if (duml.eq.2) then
          go to 151
        end if
        e=eo(1)-y/2.0
        do 69 k=1,nk
          e=e+y
          read (2,*) sa(k)
 69     continue
      end if
c     DEFINE A(I); RECALCULATE D(I) AND CALCULATE B(I)
      do 75 i=1,ni
        a(i)=0.0
        d(i)=0.0
        b(i)=0.0
        do 70 k=1,nk
          a(i)=a(i)+sa(k)*r(i,k)*y
          d(i)=d(i)+sp(k)*r(i,k)*y
 70     continue
        b(i)=a(i)/d(i)
 75   continue
c     DEFINE SIGMA(I)
      print *,'INPUT SIGMA(I) ASSUMES ALL SIGMAS ARE EQUAL'
      read *,sig
      do 80 i=1,ni
        sigma(i)=sig
 80   continue
      print *,'DO YOU WISH TO APPLY A NORMAL/GAUSSAIN NOISE'
      print *,'DISTRIBUTION TO B(I), INPUT A 1 FOR YES'
      read *,dum1
      if (dum1.eq.1) then
        print *,'INPUT THE SEED FOR THE RANDOM NUMBER'
        print *,'GENERATOR, MUST BE AN INTEGER'
        read *,dum2
        const=srand(dum2)
        do 82 i=1,ni
          sumran=0.0
          do 81 j=1,12
            const=rand(x)
            sumran=sumran+const
 81       continue
          z(i)=sigma(i)*(sumran-6.0)
          b(i)=b(i)+z(i)
          print *,'noise=',z(i)
          print *,'i=',i,'sumran=',sumran
 82     continue
      end if
      print *,'DO YOU WISH TO APPLY A NON-NEGATIVITY'
      print *,'CONSTRAINT?  INPUT 1 FOR YES'
      read *,non
      print *,'INPUT THE TYPE OF BASIS FUNCTION TO BE USED'
      print *,'TO CALCUKARE THE UNFOLDED SPECTRUM'
      print *,'     1=PLANCKIAN BLACK BODIES'
      print *,'     2=CUBIC SPLINES'
      read *,type
      if (type.eq.1) then
        print *,'INPUT THE NUMBER OF BASIS FUNCIONS TO BE'
        print *,'USED (MUST BE AN INTEGER LESS THAN 15)'
        read *,nj
c       INPUT THE INITIAL GUESS AT THE PARAMETERS OF THE
c       BASIS FUNCTIONS PROGRAM ASSUMES ONE COEFF. AND ONE
c       PARAMETER (T) PER BASIS FUNCTION
        np=2.0*nj
        do 95 j=1,nj
          print *,'INPUT THE INITIAL GUESS FOR A(J),J=',j
          read *,param(j)
 95     continue
        do 100 j=1,nj
          print *,'INPUT THE INITIAL GUESS FOR THE PARAMETER'
          print *,'T(J),J=',j,'TEMP IS IN UNITS OF Eo'
          read *,param(j+nj)
 100    continue
      end if
      if (type.eq.2) then
        print *,'INPUT THE NUMBER OF KNOTS TO BE USED TO FIT'
        print *,'THE CUBIC SPLINES (MUST BE AN INTEGER'
        print *,'6<=#KNOTS<=15) NOTE:0.0'
        print *,'IS COMPUTED AS THE FIRST KNOT, EO(1) IS'
        print *,'COMPUTED AS KNOT 2, AND 128.0*EO(1) IS'
        print *,'COMPUTED AS THE LAST KNOT.'
        print *,'INCLUDE THESE KNOTS IN THE COUNT OF YOUR'
        print *,'KNOTS AND ENSURE THE KNOTS ARE INPUT FROM'
        print *,'THE MINIMUM TO THE MAXIMUM'
        read *,nkn
        print *,'INPUT THE INITIAL GUESS AT THE KNOTS'
        print *,'KNOT 1=0.0'
        knot1=0.0
        knot2=eo(1)
        print *,'KNOT 2=',knot2
        do 101 j=1,nkn-3
          print *,'input knot ',j+2
          read *,param(j)
 101    continue
        knot15=128.0*eo(1)
        print *,'KNOT',nkn,'=',knot15
        do 102 j=1,nkn-2
          if (j.ne.1) then
            print *,'INPUT THE EXPANSION COEF. FOR'
            print *,'KNOT=',param(j-1)
          else
            print *,'INPUT THE EXPANSION COEF. FOR'
            print *,'KNOT=',knot2
          end if
          read *,param(nkn-3+j)
 102    continue
        print *,'INPUT THE TEMP. TO BE USED TO CONSTRUCT A'
        print *,'PLANCKIAN FIT BETWEEN THE LAST 2 KNOTS,'
        print *,'TEMP IS IN UNITS OF Eo'
        read *,param(nkn*2-4)
        np=nkn*2-4
      end if
c     INITIALIZE THE H MATRIX FOR THE MINIMIZATION TO THE
c     UNITY MATRIX
      do 110 p=1,np
        do 105 o=1,np
          if (o.eq.p) then
            h(o,p)=1.0
          else
            h(o,p)=0.0
          end if
 105    continue
 110  continue
c     INPUT THE INITTAL GUESS AT THE FUNCTIONAL VALUE OF THE'
c     LOWER BOUND
      print *,'INPUT THE INITIAL GUESS AT THE FUNCTIONAL'
      print *,'LOWER BOUND'
      read *,l
c     BEGIN THE APPROXIMATION OF THE TRUE SPECTRUM USING THE'
c     UNFOLDED SPECTRUM FROM THE SUBROUTINES BBFUNC AND
c     CSFUNC
```

```
      if (type.eq.1) then
        call bbfunc(param,su,y,nk,nj,pi,kc,non)
      end if
      if (type.eq.2) then
        call csfunc(param,su,nkn,nk,y,non,knot1
     *  ,knot2,knot15,eo,pi,kc)
      end if
c     CALCULATE C(I)
      do 115 i=1,ni
        c(i)=0.0
        do 111 k=1,nk
          c(i)=c(i)+su(k)*r(i,k)*y
 111    continue
 115  continue
 118  continue
c     CALCULATE CHI SQUARED
      dummy=chi2
      chi2=0.0
      do 120 i=1,ni
        chi2=chi2+((c(i)-b(i))/sigma(i))**2
 120  continue
      print *,'it=',it,'chi2=',chi2
      dum3=abs((dummy-chi2)/chi2)
      if (chi2.le.1.0e-2) then
        print *,'minimization required',it,'iterations'
        print *,'chi2=',chi2
        do 122 p=1,np
          print *,'parameter=',param(p)
 122    continue
        e=(eo(1)-y/2.0)+(nk+1.0)*y
        do 130 k=1,nk
          e=e-y
          write (7,*) e,' ',sa(nk+1-k)
 130    continue
        write (7,*) eo(1),' ','0.0'
        if (type.eq.1) then
          call bbfunc(param,su,y,nk,nj,pi,kc,non)
        end if
        if (type.eq.2) then
          call csfunc(param,su,nkn,nk,y,non
     *    ,knot1,knot2,knot15,eo,pi,kc)
        end if
        e=eo(1)-y/2.0
        do 131 k=1,nk
          e=e+y
          write (7,*) e,' ',su(k)
 131    continue
        go to 151
      end if
      if (chi2.lt.ni.and.dum3.lt.1.0e-2) then
c     PRINT ACTUAL SPECTRUM AND THE APPROXIMATED SPECTRUM
c     TO AN OUTPUT FILE
```

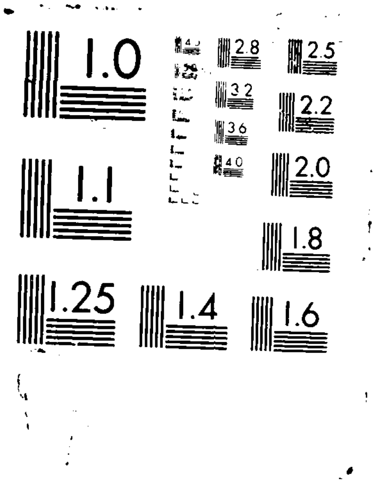
```
        print *,'minimization required',it,'iterations'
        print *,'chi2=',chi2
        do 140 p=1,np
          print *,'parameter=',param(p)
 140    continue
        e=(eo(1)-y/2.0)+(nk+1)*y
        do 149 k=1,nk
          e=e-y
          write (7,*) e,' ',sa(nk-k+1)
 149    continue
        write (7,*) eo(1),' ','0.0'
        if (type.eq.1) then
          call bbfunc(param,su,y,nk,nj,pi,kc,non)
        end if
        if (type.eq.2) then
          call csfunc(param,su,nkn,nk,y,non,
     *    knot1,knot2,knot15,eo,pi,kc)
        end if
        e=eo(1)-y/2.0
        do 150 k=1,nk
          e=e+y
          write (7,*) e,' ',su(k)
 150    continue
      else
        it=it+1
        call minimize(param,y,nk,nj,pi,kc,non,c,b,sigma
     *  ,np,ni,h,r,type,nkn,knot1,knot2,knot15,eo)
        go to 118
      end if
 151  continue
      close (7)
      stop
      end


c     PLANCKIAN BLACK BODY BASIS FUNCTION SUBROUTINE
      subroutine bbfunc(param,su,y,nk,nj,pi,kc,non)
      real*8 eo(20),x,pi,kc,param(30),su(1280),e,y
      integer k,j,nk,nj,non
      common eo(20)
      print *,eo(1)
      e=eo(1)-(y/2.0)
      do 2000 k=1,nk
        e=e+y
        su(k)=0.0
        do 1900 j=1,nj
          x=param(j)*(15.0/(pi*kc*param(j+nj))**4)
          if (e/(kc*param(j+nj)).gt.86.0) then
            su(k)=0.0
          else
            su(k)=su(k)+x*(e**3.0/(exp(e/(kc*param(j+nj)))
     *      -1.0))
```

```
          end if
          if (non.eq.1.and.su(k).lt.0.0) then
            su(k)=0.0
         end if
 1900   continue
 2000 continue
      return
      end


c     CUBIC SPLINE BASIS FUNCTION SUBROUTINE
      subroutine csfunc(param,su,nkn,nk,y,non,
     *knot1,knot2,knot15,eo,pi,kc)
      real*8 param(30),kn(15),coef(15),num,den
      real*8 planck,const,pi,kc,t,x,slope,su(1280),y,e
      real*8 knot1,knot2,knot15,eo(20)
      integer j,k,non,nk,nkn
      kn(1)=knot1
      kn(2)=knot2
      do 2010 j=3,nkn-1
        kn(j)=param(j-2)
 2010 continue
      kn(nkn)=knot15
      do 2020 j=2,nkn-1
        coef(j)=param(j+nkn-4)
 2020 continue
      t=param(2*nkn-4)
      e=eo(1)-y/2.0
      do 2035 k=1,nk
        e=e+y
        su(k)=0.0
        do 2030 j=2,nkn
          if (j.eq.2) then
            if (e.ge.kn(3).and.e.lt.kn(4)) then
              num=(e-kn(j+1))*(e-kn(j+2))*(e-kn(j+3))
              den=(kn(j)-kn(j+1))*(kn(j)-kn(j+2))*
     *        (kn(j)-kn(j+3))
              su(k)=coef(j)*num/den+su(k)
            end if
            if (e.ge.kn(2).and.e.lt.kn(3)) then
              slope=1.0/(kn(2)-kn(3))
              su(k)=coef(j)*(1.0+((e-kn(2))*slope))+su(k)
            end if
            if (e.lt.kn(2).or.e.ge.kn(4)) then
              su(k)=0.0+su(k)
            end if
          end if
          if (j.eq.3) then
            if (e.ge.kn(4).and.e.lt.kn(5)) then
              if (nkn.gt.6) then
                num=(e-kn(j+1))*(e-kn(j+2))*(e-kn(j+3))
                den=(kn(j)-kn(j+1))*(kn(j)-kn(j+2))*
     *          (kn(j)-kn(j+3))
                su(k)=coef(j)*num/den+su(k)
              else
                su(k)=0.0+su(k)
              end if
            end if
            if (e.gt.kn(3).and.e.lt.kn(4)) then
              num=(e-kn(j-1))*(e-kn(j+1))*(e-kn(j+2))
              den=(kn(j)-kn(j-1))*(kn(j)-kn(j+1))*
     *        (kn(j)-kn(j+2))
              su(k)=coef(j)*num/den+su(k)
            end if
            if (e.ge.kn(2).and.e.lt.kn(3)) then
              slope=1.0/(kn(3)-kn(2))
              su(k)=coef(j)*((e-kn(2))*slope)+su(k)
            end if
            if (e.lt.kn(2).or.e.ge.kn(5)) then
              su(k)=0.0+su(k)
            end if
          end if
          if (j.ge.4.and.j.lt.(nkn-2)) then
            if (e.ge.kn(j+1).and.e.lt.kn(j+2)) then
              if (kn(j+2).ne.kn(nkn-1)) then
                num=(e-kn(j+1))*(e-kn(j+2))*(e-kn(j+3))
                den=(kn(j)-kn(j+1))*(kn(j)-kn(j+2))*
     *          (kn(j)-kn(j+3))
                su(k)=coef(j)*num/den+su(k)
              else
                su(k)=0.0+su(k)
              end if
            end if
            if (e.ge.kn(j).and.e.lt.kn(j+1)) then
              num=(e-kn(j-1))*(e-kn(j+1))*(e-kn(j+2))
              den=(kn(j)-kn(j-1))*(kn(j)-kn(j+1))*
     *        (kn(j)-kn(j+2))
              su(k)=coef(j)*num/den+su(k)
            end if
            if (e.ge.kn(j-1).and.e.lt.kn(j)) then
              num=(e-kn(j-2))*(e-kn(j-1))*(e-kn(j+1))
              den=(kn(j)-kn(j-2))*(kn(j)-kn(j-1))*
     *        (kn(j)-kn(j+1))
              su(k)=coef(j)*num/den+su(k)
            end if
            if (e.ge.kn(j-2).and.e.lt.kn(j-1)) then
              if (kn(j-2).ne.kn(2)) then
                num=(e-kn(j-3))*(e-kn(j-2))*(e-kn(j-1))
                den=(kn(j)-kn(j-3))*(kn(j)-kn(j-2))*
     *          (kn(j)-kn(j-1))
                su(k)=coef(j)*num/den+su(k)
              else
                su(k)=0.0+su(k)
              end if
            end if
            if (e.lt.kn(j-2).or.e.ge.kn(j+2)) then
              su(k)=0.0+su(k)
            end if
          end if
          if (j.eq.(nkn-2)) then
            if (e.gt.kn(nkn-2).and.e.lt.kn(nkn-1)) then
              slope=1.0/(kn(nkn-2)-kn(nkn-1))
              su(k)=coef(j)*(1.0+((e-kn(nkn-2))*slope))+su(k)
            end if
            if (e.ge.kn(nkn-3).and.e.lt.kn(nkn-2)) then
              num=(e-kn(j-2))*(e-kn(j-1))*(e-kn(j+1))
              den=(kn(j)-kn(j-2))*(kn(j)-kn(j-1))*
     *        (kn(j)-kn(j+1))
              su(k)=coef(j)*num/den+su(k)
            end if
            if (e.ge.kn(nkn-4).and.e.lt.kn(nkn-3)) then
              num=(e-kn(j-3))*(e-kn(j-2))*(e-kn(j-1))
              den=(kn(j)-kn(j-3))*(kn(j)-kn(j-2))*
     *        (kn(j)-kn(j-1))
              su(k)=coef(j)*num/den+su(k)
            end if
            if (e.lt.kn(nkn-4).or.e.ge.kn(nkn-1)) then
              su(k)=0.0+su(k)
            end if
          end if
          if (j.eq.(nkn-1)) then
            if (e.ge.kn(nkn-2).and.e.lt.kn(nkn-1)) then
              slope=1.0/(kn(nkn-1)-kn(nkn-2))
              su(k)=coef(j)*((e-kn(nkn-2))*slope)+su(k)
            end if
            if (e.ge.kn(nkn-3).and.e.lt.kn(nkn-2)) then
              num=(e-kn(j-3))*(e-kn(j-2))*(e-kn(j-1))
              den=(kn(j)-kn(j-3))*(kn(j)-kn(j-2))*
     *        (kn(j)-kn(j-1))
              su(k)=coef(j)*num/den+su(k)
            end if
            if (e.lt.kn(nkn-3).or.e.ge.kn(nkn-1)) then
              su(k)=0.0+su(k)
            end if
          end if
          if (j.eq.nkn) then
            if (e.ge.kn(j-1).and.e.le.kn(j)) then
              x=15.0/(pi*kc*t)**4
              planck=x*(kn(j-1)**3.0)/
     *        (exp(kn(j-1)/(kc*t))-1.0)
              const=coef(j-1)/planck
              if (e/(kc*t).ge.69.0) then
                su(k)=0.0+su(k)
              else
                su(k)=const*x*(e**3.0)/
```

AD-A190 577 AN APPROXIMATION TECHNIQUE FOR SOLVING A SYSTEM OF 2/2
FREDHOLM INTEGRAL EQUA..(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. R B AANIEL
UNCLASSIFIED MAR 88 AFIT/GNE/ENG/88M-3 F/G 12/5 NL


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

```
     *          (exp(e/(kc*t))-1.0)+su(k)
              end if
            else
              su(k)=0.0+su(k)
            end if
          end if
 2030   continue
        if (non.eq.1.and.su(k).lt.0.0) then
          su(k)=0.0
        end if
 2035 continue
      return
      end



c     SUBROUTINE TO MINIMIZE CHI SQUARED
      subroutine minimize(param,y,nk,nj,pi,kc,non,c
     *,b,sigma,np,ni,h,r,type,nkn,knot1,knot2,knot15,eo)
      real*8 grchi2(30),param(30),b(20),c(20),sigma(20)
      real*8 gradc(30),sprime(30),pprime,t
      real*8 y,pi,kc,r(20,1280),h(30,30),norms,s(30)
      real*8 v(30,1),gprime(30),g(30),q(30,1),a(30,30)
      real*8 duml,sumi,sumj
      real*8 sumk,su(1280),z(30,30),sump,sumo,pc(30),l
      real*8 kn(15),knot1,knot2,knot15,eo(20)
      integer nkn,p,np,i,ni,o,reset,nk,nj,non,j,k,type
c     CALCULATE THE GRADIENT OF CHI SQUARED
      t=0
      do 2500 p=1,np
        grchi2(p)=0.0
        call gradientc(param,y,nk,nj,pi,kc,non,c,
     *  gradc,r,np,ni,p,type,nkn,knot1,knot2,knot15,eo)
        do 2400 i=1,ni
          grchi2(p)=grchi2(p)+((c(i)-b(i))/(sigma(i)**2))
     *    *gradc(i)
 2400   continue
        grchi2(p)=grchi2(p)*2.0
 2500 continue
 2525 continue
c     CALCULATE THE SEARCH DIRECTION
      do 2600 p=1,np
        sprime(p)=0.0
        do 2550 o=1,np
          sprime(p)=-h(p,o)*grchi2(o)+sprime(p)
 2550   continue
 2600 continue
c     NORMALIZE TO FIND THE UNIT SEARCH DIRECTION
      duml=0.0
      do 2700 p=1,np
        duml=duml+sprime(p)**2.0
 2700 continue
```

```
          norms=sqrt(duml)
        do 2800 p=1,np
          s(p)=sprime(p)/norms
 2800 continue
        call linsearch(np,param,t,s,y,nk,nj,pi,kc,non,ni
       *,r,b,sigma,h,reset,l,type,nkn,knot1,knot2,knot15,eo)
        if (reset.eq.1) then
          reset=0.0
          do 2820 p=1,np
            do 2810 o=1,np
              if (o.eq.p) then
                h(p,o)=1.0
              else
                h(p,o)=0.0
              end if
 2810       continue
 2820     continue
          go to 2525
        end if
 2850 continue
        do 2900 p=1,np
          pprime(p)=param(p)+t*s(p)
 2900 continue
        if (type.eq.2) then
          do 2903 p=1,nkn-3
            do 2902 j=1,nkn-3
              kn(j+2)=pprime(j)
 2902       continue
            kn(1)=knot1
            kn(2)=knot2
            kn(nkn)=knot15
            if (kn(p+2).le.kn(p+1).or.kn(p+2).ge.kn(p+3)) then
              t=t/2.0
              go to 2850
            end if
 2903     continue
        end if
        do 2904 p=1,np
          print *,'parameter=',param(p),'s)p)=',s(p),'t=',t
 2904 continue
c       CALCULATE NEW H MATRIX BASED ON NEW PARAMETERS
        do 2905 p=1,np
          v(p,1)=pprime(p)-param(p)
 2905 continue
c       CALCULATE A NEW C(I) FOR NEW PARAMETERS
        if (type.eq.1) then
          call bbfunc(pprime,su,y,nk,nj,pi,kc,non)
        end if
        if (type.eq.2) then
          call csfunc(pprime,su,nkn,nk,y,non,knot1
       *    ,knot2,knot15,eo,pi,kc)
        end if
```

```
      do 2908 i=1,ni
        pc(i)=0.0
        do 2907 k=1,nk
          pc(i)=pc(i)+su(k)*r(i,k)*y
2907    continue
2908  continue
      do 2915 p=1,np
        call gradientc(pprime,y,nk,nj,pi,kc,non,pc
     *  ,gradc,r,np,ni,p,type,nkn,knot1,knot2,knot15,eo)
        gprime(p)=0.0
        do 2910 i=1,ni
          gprime(p)=gprime(p)+((pc(i)-b(i))/sigma(i)**2)
     *    *gradc(i)
2910    continue
        gprime(p)=gprime(p)*2.0
2915  continue
      do 2925 p=1,np
        call gradientc(param,y,nk,nj,pi,kc,non,c
     *  ,gradc,r,np,ni,p,type,nkn,knot1,knot2,knot15,eo)
        g(p)=0.0
        do 2920 i=1,ni
          g(p)=g(p)+((c(i)-b(i))/sigma(i)**2)*gradc(i)
2920    continue
        g(p)=g(p)*2.0
2925  continue
      do 2930 p=1,np
        q(p,1)=gprime(p)-g(p)
2930  continue
      do 2936 p=1,np
        do 2934 o=1,np
          sumj=0.0
          do 2932 j=1,1
            sumj=sumj+v(p,j)*v(o,j)
2932      continue
          a(p,o)=sumj
2934    continue
2936  continue
      do 2942 j=1,1
        do 2940 i=1,1
          sump=0.0
          do 2938 p=1,np
            sump=sump+v(p,j)*q(p,i)
2938      continue
          duml=sump
2940    continue
2942  continue
      do 2946 p=1,np
        do 2944 o=1,np
          a(p,o)=a(p,o)/duml
2944    continue
2946  continue
      do 2956 p=1,np
```

```
          do 2954 o=1,np
            sumj=0.0
            do 2952 j=1,np
              sumi=0.0
              do 2950 i=1,l
                sumk=0.0
                do 2948 k=1,np
                  sumk=sumk+q(k,i)*h(k,o)
 2948           continue
                sumi=sumi+sumk*q(j,i)
 2950         continue
              sumj=sumj+sumi*h(p,j)
 2952       continue
            z(p,o)=sumj
 2954     continue
 2956   continue
        do 2964 i=1,l
          do 2962 j=1,l
            sump=0.0
            do 2960 p=1,np
              sumo=0.0
              do 2958 o=1,np
                sumo=sumo+h(p,o)*q(o,j)
 2958         continue
              sump=sump+sumo*q(p,i)
 2960       continue
            duml=sump
 2962     continue
 2964   continue
        do 2995 p=1,np
          do 2993 o=1,np
            z(p,o)=z(p,o)/duml
            h(p,o)=h(p,o)+a(p,o)-z(p,o)
 2993     continue
 2995   continue
c       LET PPRIME EQUAL THE NEW PARAMETERS AND RE-EVALUATE CHI
c       SQUARED
        do 3000 p=1,np
          param(p)=pprime(p)
 3000   continue
c       LET C(I)=PC(I)
        do 3010 i=1,ni
          c(i)=pc(i)
 3010   continue
        return
        end


c       SUBROUTINE TO CALCULATE THE STEP SIZE IN THE UNIT
c       SEARCH DIRECTION
        subroutine linsearch(np,param,t,s,y,nk,nj,pi,kc,non
       *,ni,r,b,sigma,h,reset,l,type,nkn,knot1,knot2,knot15,eo)
```

```
      real*8 h(30,30),h2,ht,m,t,to,ho,mo,t2,l,m2,z,duml
      real*8 tl,w,mu,sigma(20)
      real*8 param(30),s(30),y,pi,kc,r(20,1280),b(20)
      real*8 knotl,knot2,knot15,eo(20)
      integer nkn,type,np,reset,ni,nj,nk,non
      t=0.0
      call calcfunc(np,param,t,s,y,nk,nj,pi,kc,non
     *,ni,r,ht,b,sigma,m,type,nkn,knotl,knot2,knot15,eo)
      ho=ht
      mo=m
      if (mo.gt.0.0) then
        reset=1
        return
      end if
      to=0.0
      if(ho.le.1) then
        l=ho+(mo/2.0)
      end if
      if (1.0.lt.(-2.0*(ho-l)/mo)) then
        t2=1.0
      else
        t2=-2.0*(ho-l)/mo
      end if
 3600 continue
      t=t2
      call calcfunc(np,param,t,s,y,nk,nj,pi,kc,non
     *,ni,r,ht,b,sigma,m,type,nkn,knotl,knot2,knot15,eo)
      h2=ht
      m2=m
      if (m2.gt.0.0) then
        z=3.0*(ho-h2)/(t2-to)+mo+m2
        duml=z*2.0+mo+m2
        if(duml.eq.0.0) then
          tl=to+mo*(t2-to)/(2.0*(z+mo))
        else
          w=sqrt(z**2-mo*m2)
          mu=(m2+w-z)/(2.0*w+m2-mo)
          tl=t2-mu*(t2-to)
        end if
      end if
      if(m2.le.0.0.and.h2.gt.ho) then
        t2=(to+t2)/2.0
        go to 3600
      end if
      if (m2.eq.0.0.and.h2.lt.ho) then
        tl=t2
      end if
      if (m2.lt.0.0.and.h2.le.ho) then
        duml=t2-to
        to=t2
        ho=h2
        mo=m2
```

```
          t2=t2+2.0*(duml)
          go to 3600
        end if
        if (m2.eq.0.0.and.ho.eq.h2) then
          tl=to
        end if
        t=tl
        return
        end


c       SUBROUTINE TO CALCULATE THE FUNCTION TO MINIMIZE
c       H=F(PARAM+T*S)
        subroutine calcfunc(np,param,t,s,y,nk,nj,pi,kc
       *,non,ni,r,ht,b,sigma,m,type,nkn,knotl,knot2,knotl5,eo)
        real*8 y,tparam(30),param(30),su(1280),b(20),tc(30)
        real*8 t,gradc(30),m,s(30),e,ht,sigma(20)
        real*8 grchi2(30),pi,kc,r(20,1280)
        real*8 knotl,knot2,knotl5,eo(20)
        integer nkn,type,p,i,np,ni,nk,k,non,nj
        do 4000 p=1,np
         tparam(p)=param(p)+t*s(p)
 4000   continue
c       CALCULATE THE UNFOLDED SPECTRUM
        if (type.eq.1) then
          call bbfunc(tparam,su,y,nk,nj,pi,kc,non)
        end if
        if (type.eq.2) then
          call csfunc(tparam,su,nkn,nk,y,non,knotl
       *  ,knot2,knotl5,eo,pi,kc)
        end if
c       CALCULATE C(I) FOR NEW PARAMETERS
        do 4200 i=1,ni
          tc(i)=0.0
          do 4100 k=1,nk
            tc(i)=tc(i)+su(k)*r(i,k)*y
 4100     continue
 4200   continue
c       CALCULATE CHI SQUARED FOR THE NEW PARAMETERS
        ht=0.0
        do 4300 i=1,ni
          ht=ht+((tc(i)-b(i))/sigma(i))**2.0
 4300   continue
c       CALCULATE THE SLOPE OF CHI SQUARED
        do 4500 p=1,np
          call gradientc(tparam,y,nk,nj,pi,kc,non
       *  ,tc,gradc,r,np,ni,p,type,nkn,knotl,knot2,knotl5,eo)
          grchi2(p)=0.0
          do 4400 i=1,ni
            grchi2(p)=grchi2(p)+((tc(i)-b(i))/sigma(i)**2.0)
       *    *gradc(i)
 4400   continue
          grchi2(p)=2.0*grchi2(p)
 4500 continue
      m=0.0
      do 4600 p=1,np
        m=m+grchi2(p)*s(p)
 4600 continue
      return
      end



c     SUBROUTINE TO CALCULATE THE GRADIENT OF C(I)
      subroutine gradientc(param,y,nk,nj,pi,kc,non,c,gradc
     *,r,np,ni,x,type,nkn,knot1,knot2,knot15,eo)
      real*8 delta,newpar(30),param(30),f(30),su(1280)
      real*8 r(20,1280),gradc(30),c(20)
      real*8 knot1,knot2,knot15,eo(20),pi,y,kc
      integer nkn,i,x,ni,p,np,k,nk,nj,non,type
      delta=0.01
      do 5000 p=1,np
        if (p.eq.x) then
          newpar(p)=param(p)+delta*param(p)
        else
          newpar(p)=param(p)
        end if
 5000 continue
      if (type.eq.1) then
        call bbfunc(newpar,su,y,nk,nj,pi,kc,non)
      end if
      if (type.eq.2) then
        call csfunc(newpar,su,nkn,nk,y,non,knot1
     *  ,knot2,knot15,eo,pi,kc)
      end if
c     CALCULATE C(I) FOR THE NEW PARAMETERS (LET IT EQUAL
c     F(I))
      do 5200 i=1,ni
        f(i)=0.0
        do 5100 k=1,nk
          f(i)=f(i)+su(k)*r(i,k)*y
 5100   continue
 5200 continue
      do 5300 i=1,ni
        gradc(i)=(f(i)-c(i))/(delta*param(x))
        if (abs(f(i)-c(i)).lt.1e-6) then
          print *,'WARNING F(I)-C(I) LESS THAN 1.0E-6'
        end if
 5300 continue
      return
      end
```

Appendix E: Validation Results

This appendix is a continuation of the results presented in Section V. The cases studied using one planckian basis function are presented in Table XII and the final four cases studied using cubic spline basis functions with fixed knots are presented in Table XIII. Additionally, the linear plots of the actual and unfolded spectra for the BC cases are presented. The actual and unfolded spectra for the BP and BF1, BF2, BF3, and BF4 were identical so no plot is presented. Also, the plots for cases BF5, BF6, BF7, and BF8 are presented even though χ^2 was greater than 20.0.

TABLE XII
Continuation of the Planckian Basis
Function Validation Test (a,b)

Parameters			
Initial		Unfolded	
a	T	a	T
2.0	5.0	2.0	5.0
3.0	3.0	2.0	5.0
3.0	4.0	2.0	5.0
1.0	6.0	2.0	5.0
0.5	6.0	2.0	5.0
1.0	2.0	2.0	5.0
1.0	4.0	2.0	5.0
4.0	7.0	2.0	5.0

a. Actual parameters equal 2.0 and 5.0

b. $\sigma_i = 0.01$ and $\chi^2 \leq 0.01$ in all cases The initial case is the initial guess at the unfolded spectrum

TABLE XIII

Continuation of the Validation Cases for the Cubic Spline Basis Functions With Fixed Knots (a,b,c)

Case	Spectra	Knots		Coefficients			T	χ^2
BF5	Actual	10.0	25.0	4.0	5.0	3.0	10.0	
		50.0	80.0	1.6	1.0			
	Initial	5.0	30.0	2.0	7.0	1.0	2.0	
		40.0	60.0	3.0	4.0			
	Unfolded	5.0	30.0	3.9	4.6	2.5	14.0	33.0
		40.0	60.0	1.9	1.5			
BF6	Actual	5.0	10.0	4.0	5.0	3.0	10.0	
		25.0	50.0	1.6	1.0			
	Initial	7.0	30.0	2.0	7.0	1.0	20.0	
		70.0	90.0	3.0	4.0			
	Unfolded	7.0	30.0	4.6	3.8	-0.3	2.0	2.3e4
		70.0	90.0	-17.0	-81.0			
BF7	Actual	5.0	10.0	4.0	5.0	3.0	10.0	
		25.0	50.0	1.6	1.0			
	Initial	7.0	30.0	2.0	7.0	1.0	2.0	
		70.0	90.0	3.0	4.0			
	Unfolded	7.0	30.0	5.0	3.5	1.5	-12.0	1.9e3
		70.0	90.0	0.55	-0.17			
BF8	Actual	10.0	25.0	4.0	5.0	3.0	10.0	
		50.0	80.0	1.6	1.0			
	Initial(c)	5.0	30.0	2.0	7.0	1.0	20.0	
		40.0	60.0	3.0	4.0			
	Unfolded	5.0	30.0	3.9	4.6	2.5	14.0	33.0
		40.0	60.0	1.9	1.5			

a. $\sigma_i = 0.01$

b. Convergence criteria: $\chi^2 \leq 1.0e\text{-}2$ for all cases or 0.01% change in χ^2 for case BF5 and 0.1% change in χ^2 for cases BF6, BF7, BF8

c. Fixed knots at 0.0,1.0,128.0 are implicit in definition of these splines.

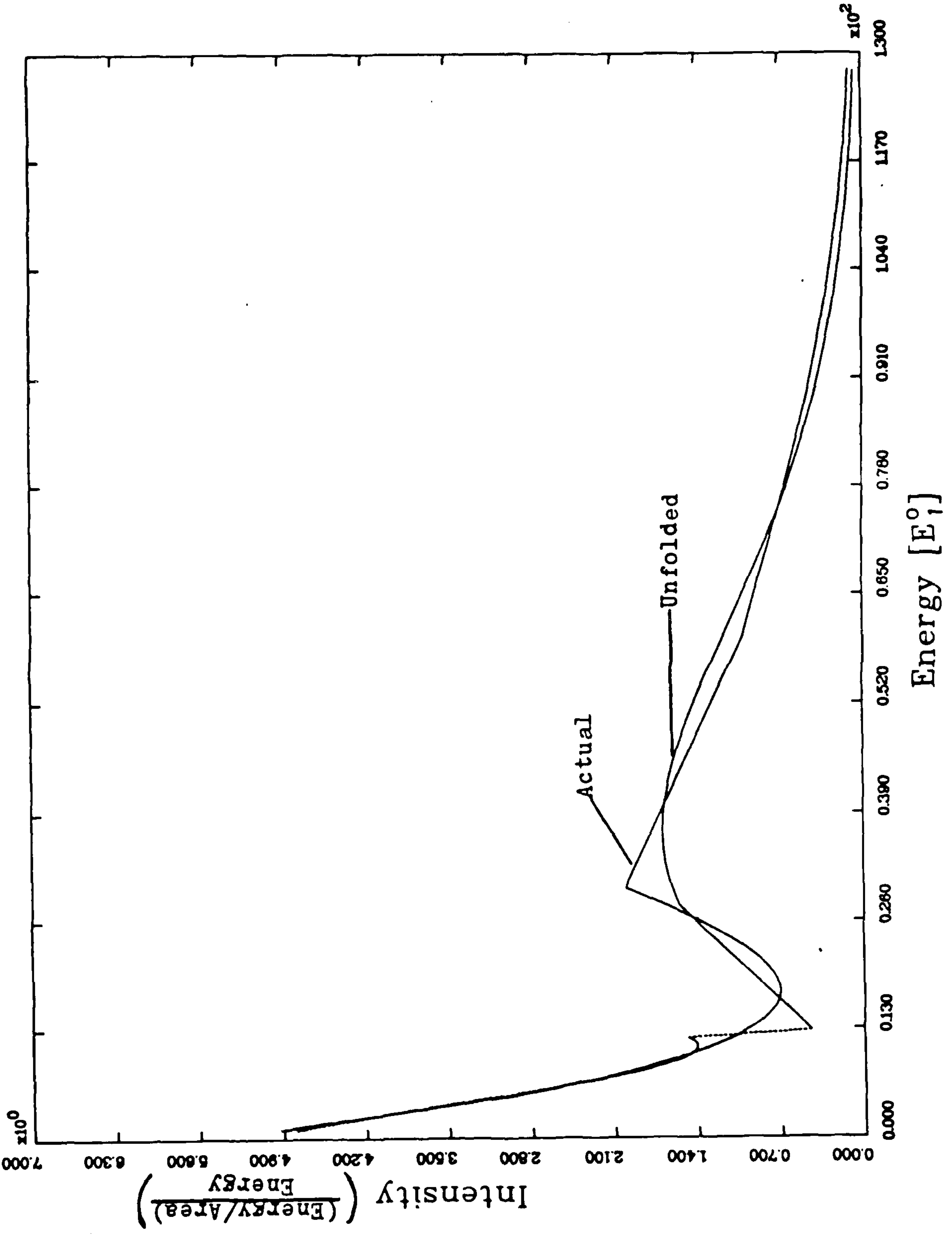


Figure 13. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case BC1

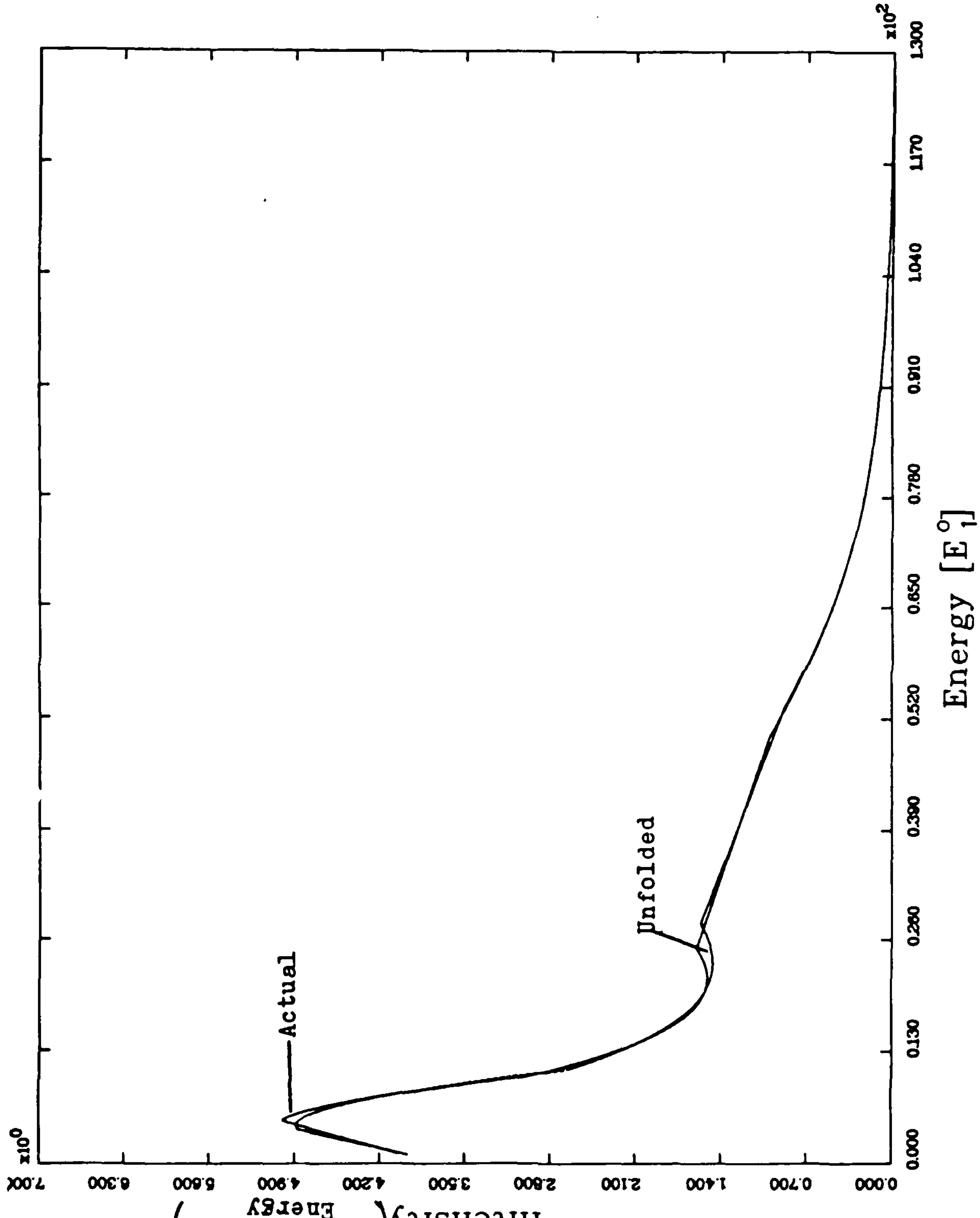


Figure 14. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case BC2

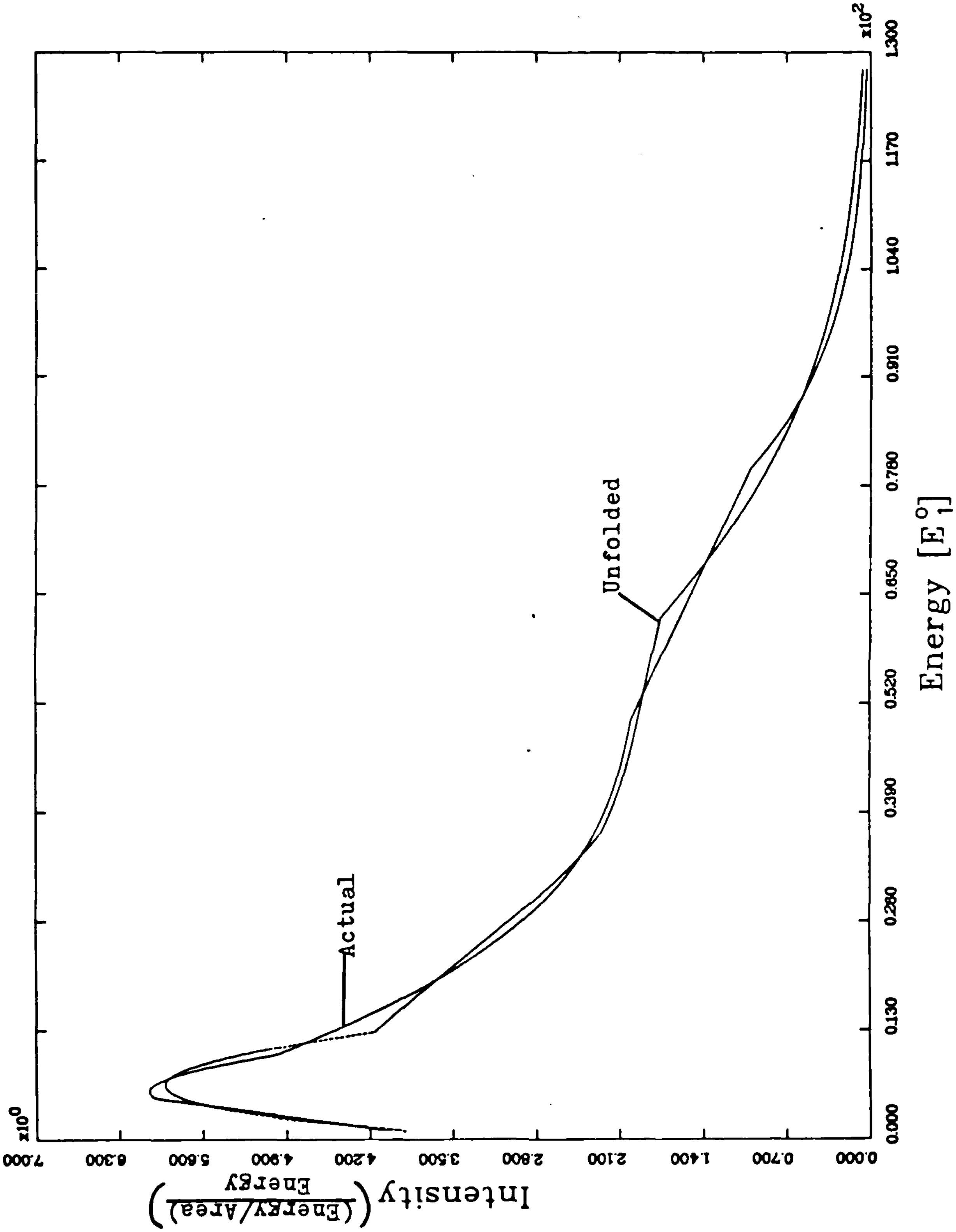


Figure 15. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case BC3

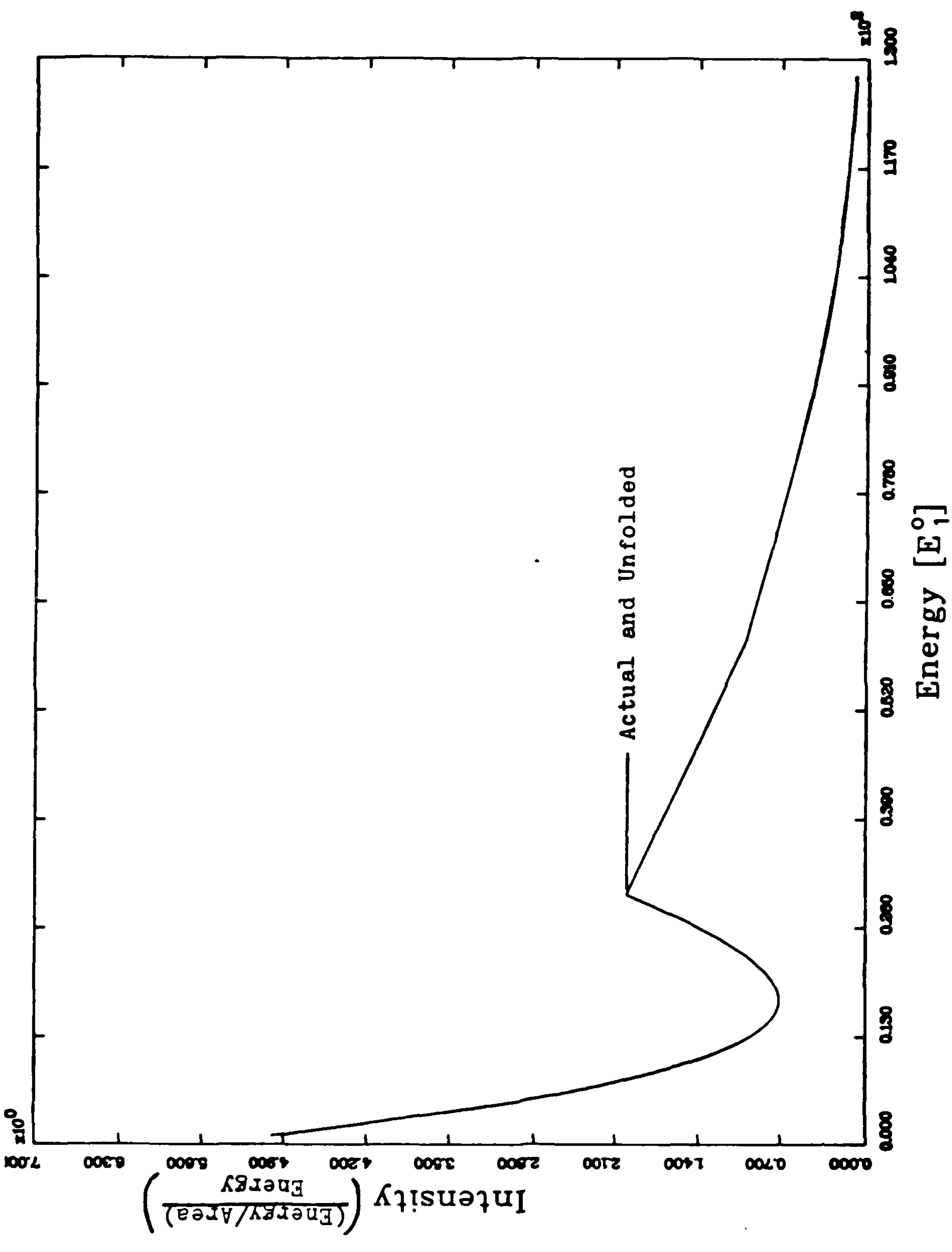


Figure 16. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case BC4

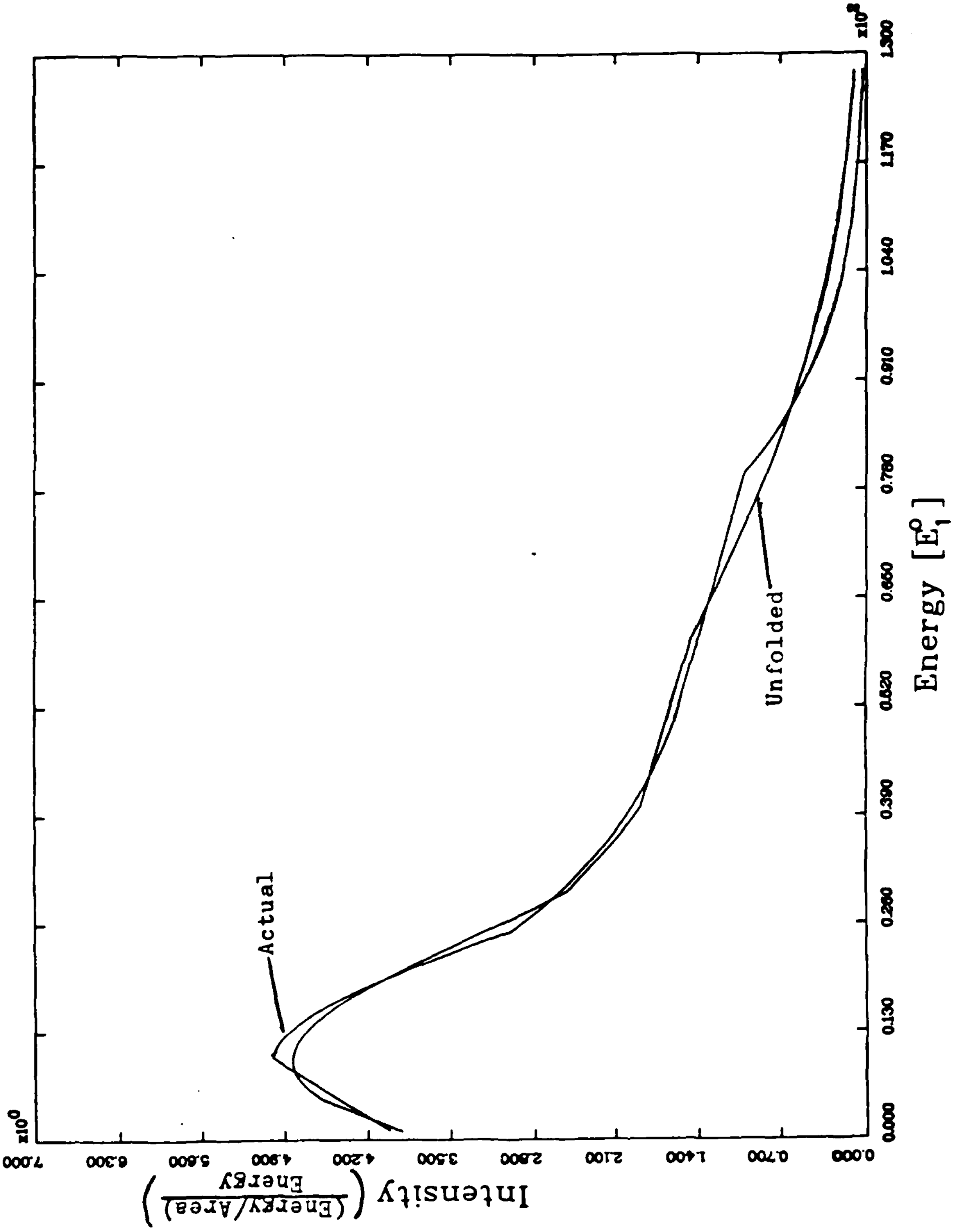


Figure 17. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case BF5

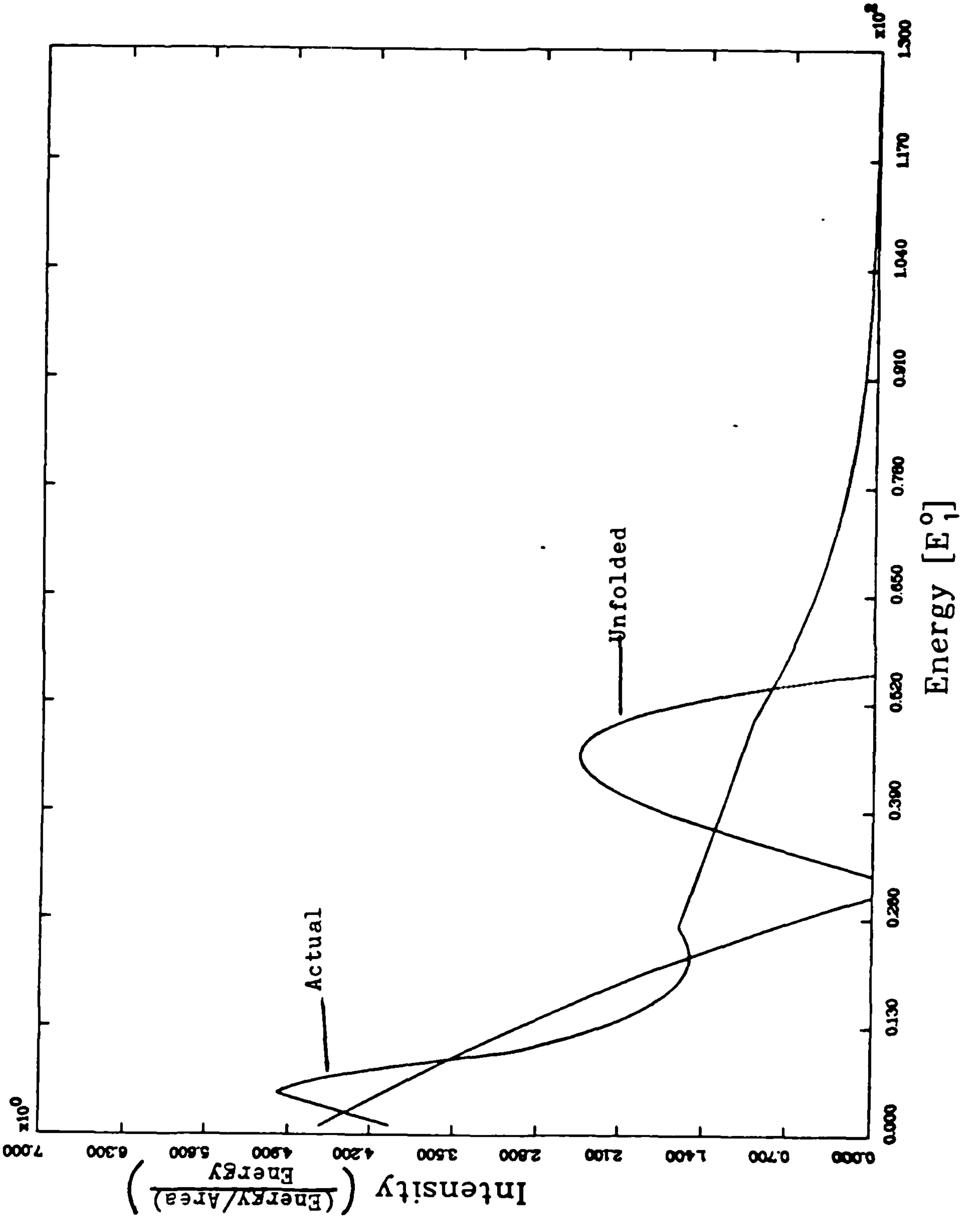


Figure 18. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case BF6

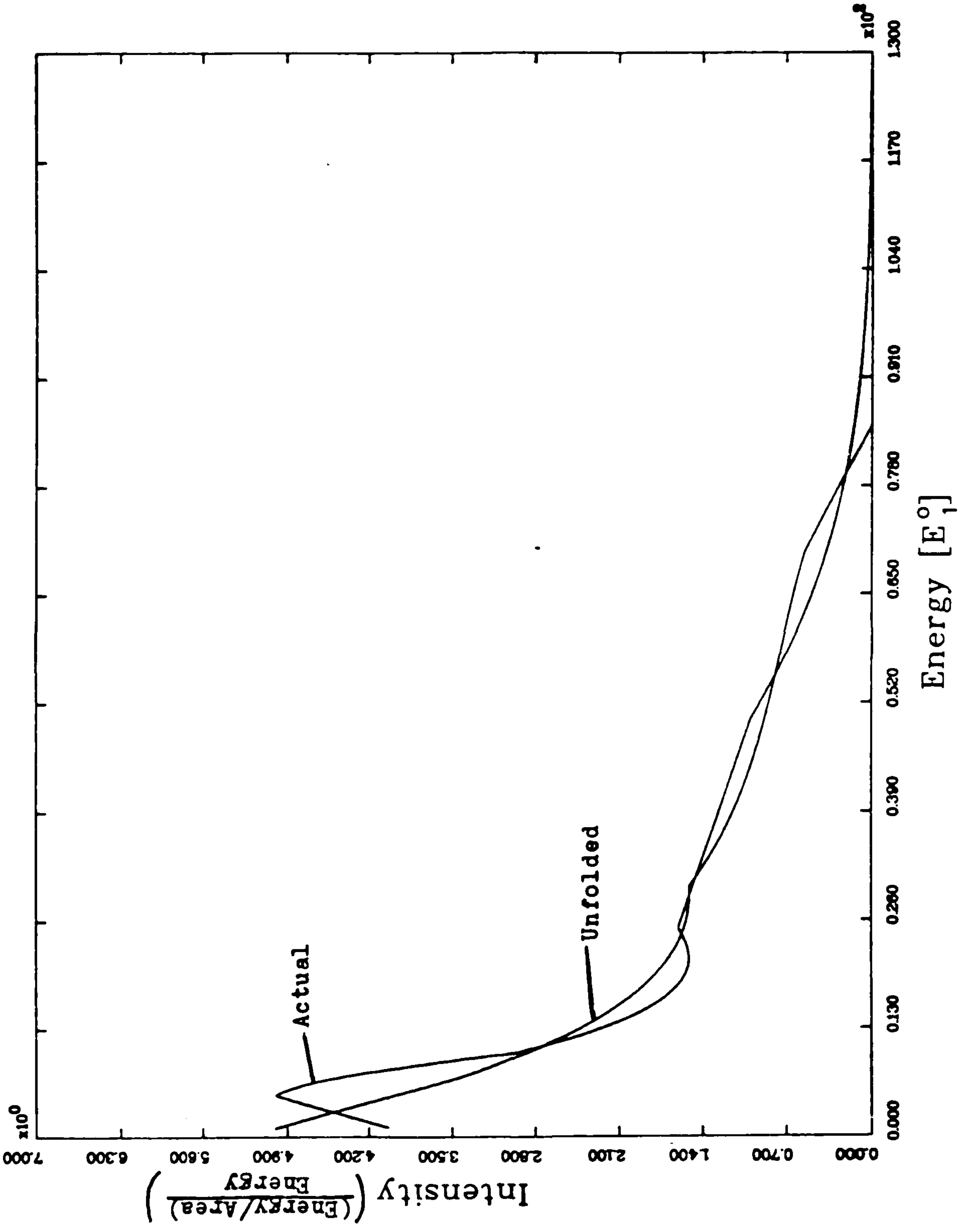


Figure 19. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case BF7

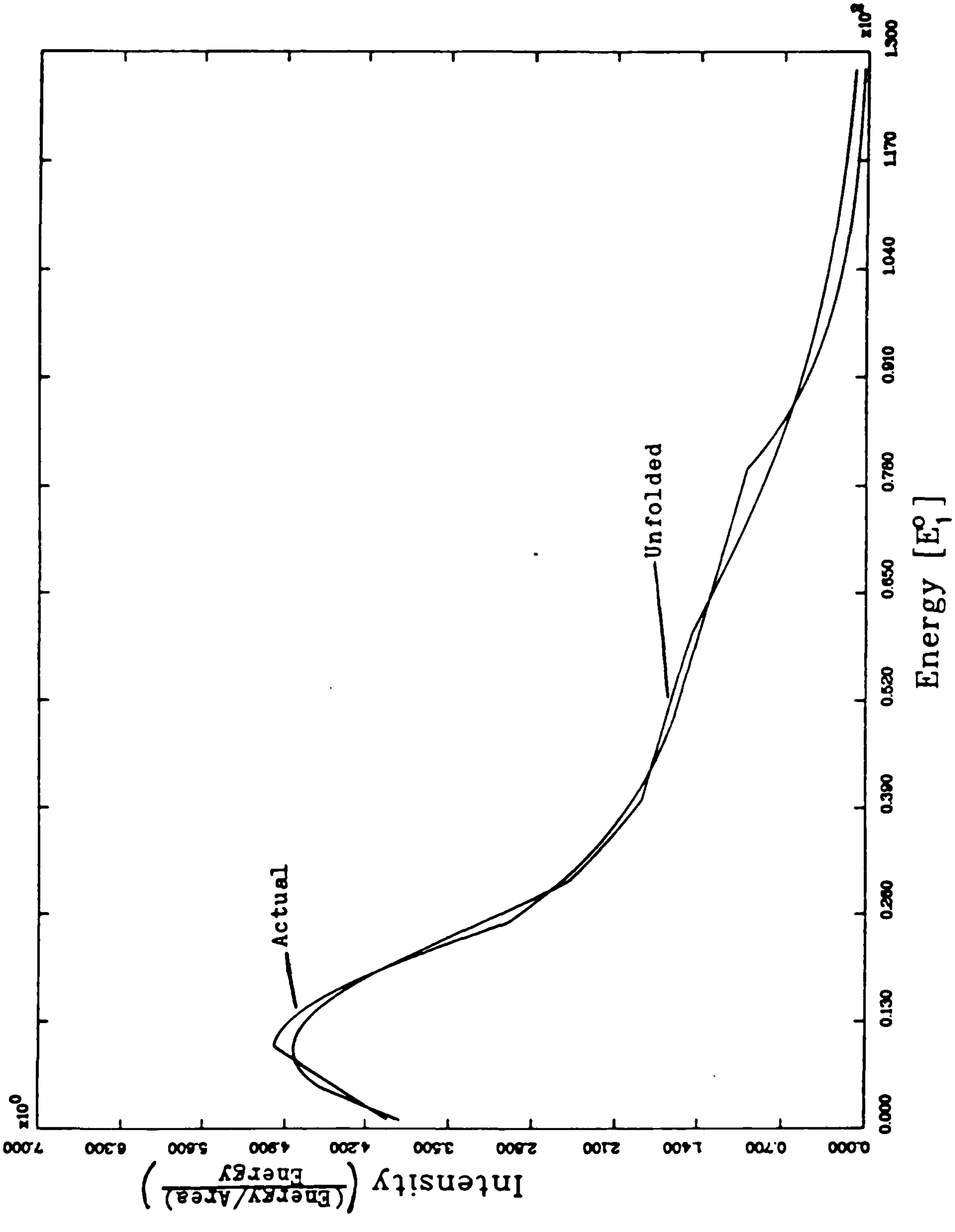


Figure 20. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case BF8

Appendix F: Test Case Results

This appendix is a continuation of Section V and presents the linear plots of the cases presented in Section V. If the linear plot of the actual and unfolded spectra for a given test case is not in this section and the χ^2 value for the case in Section V is acceptable (i.e. less than 20.0), then either the actual and unfolded spectra match exactly or the plots were presented in Section V.

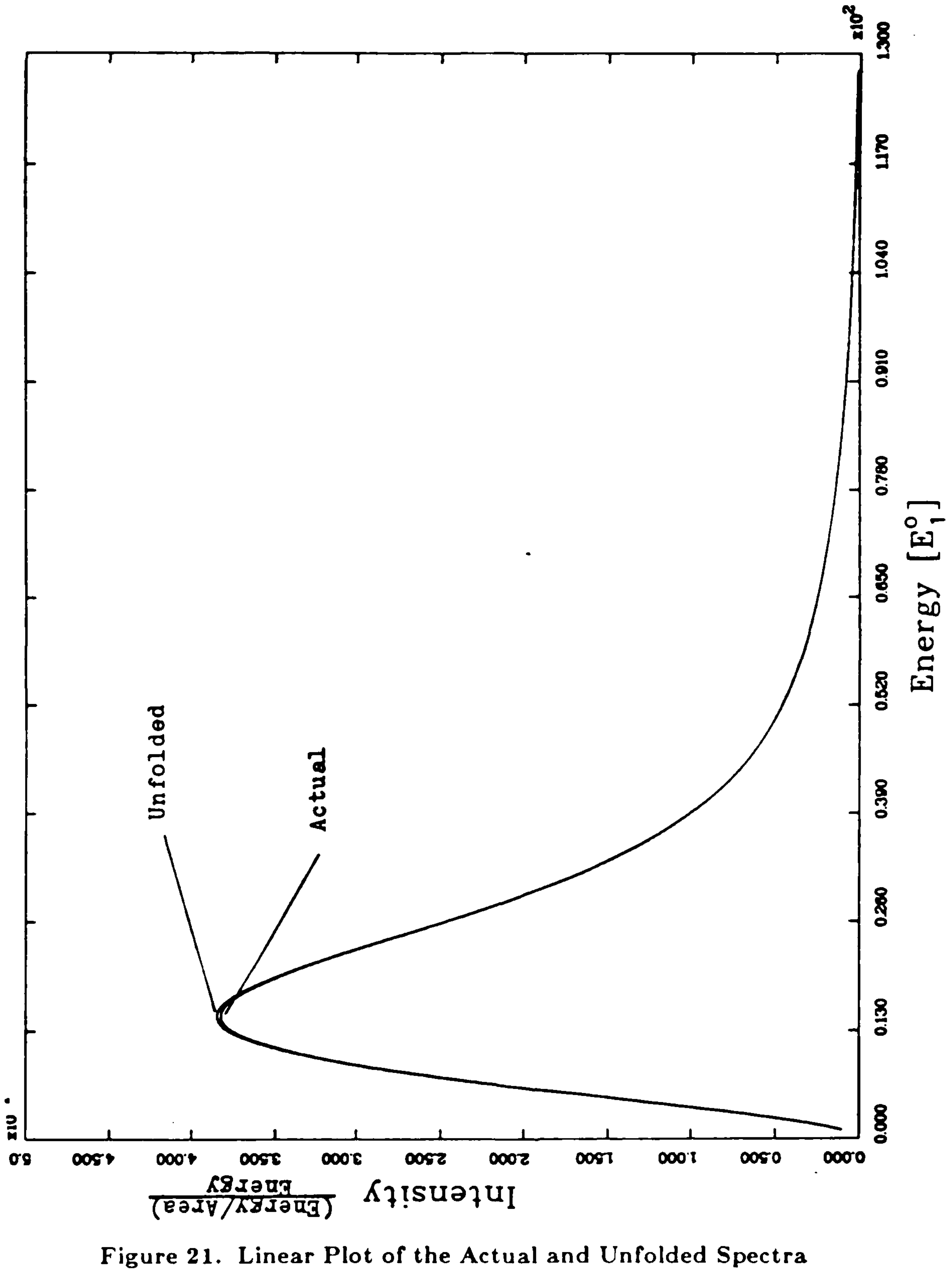


Figure 21. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP0

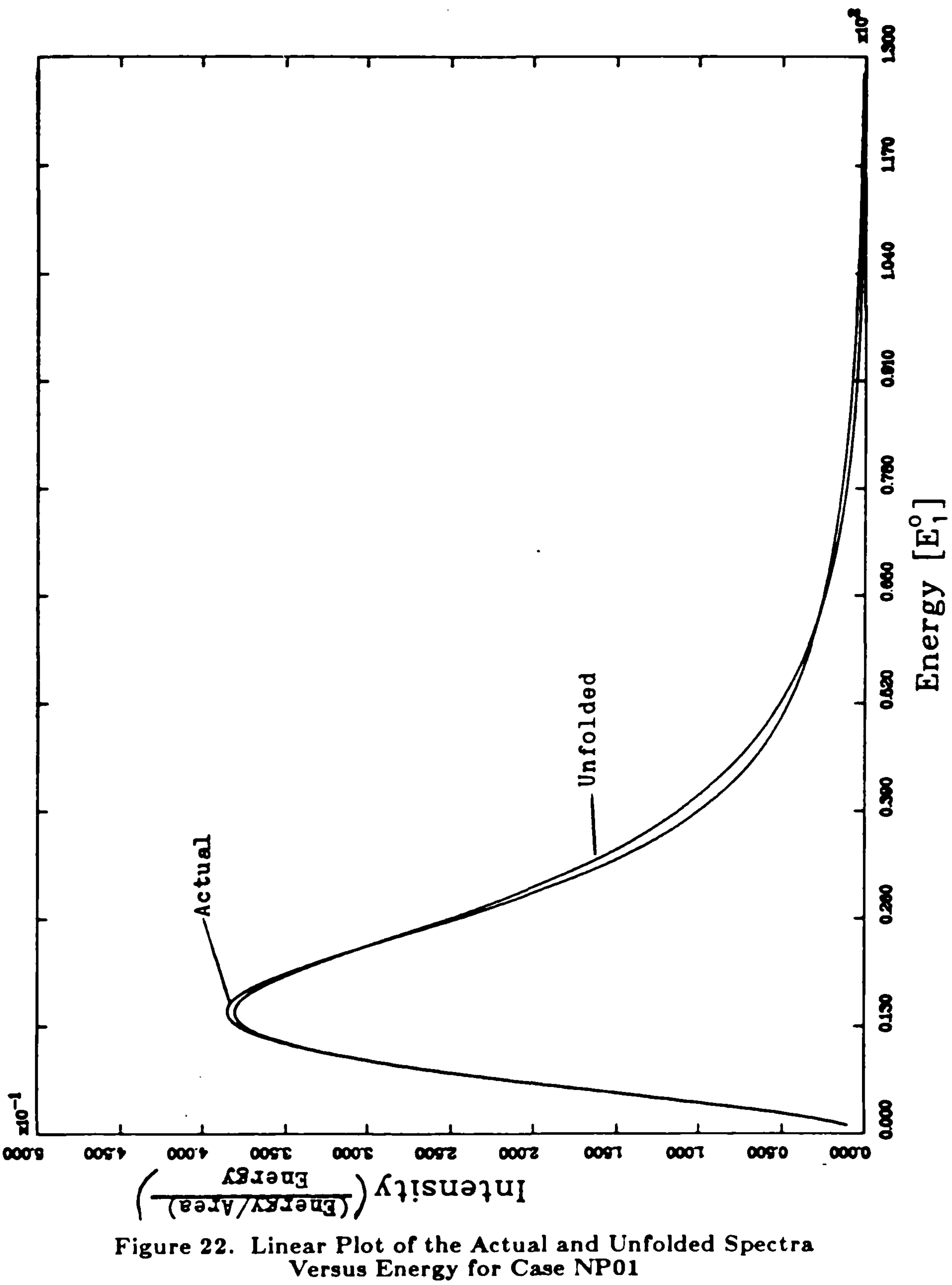


Figure 22. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP01

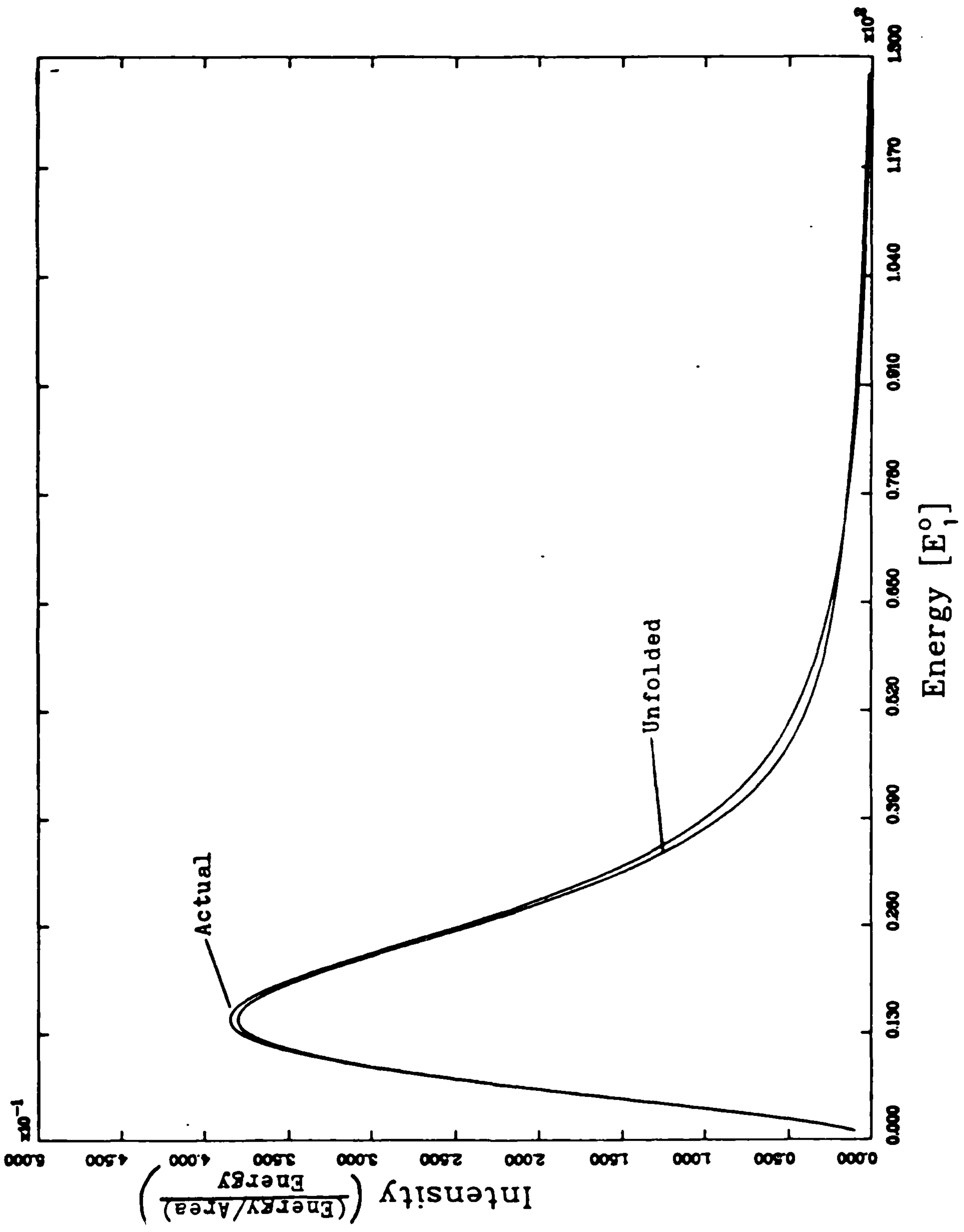


Figure 23. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP02

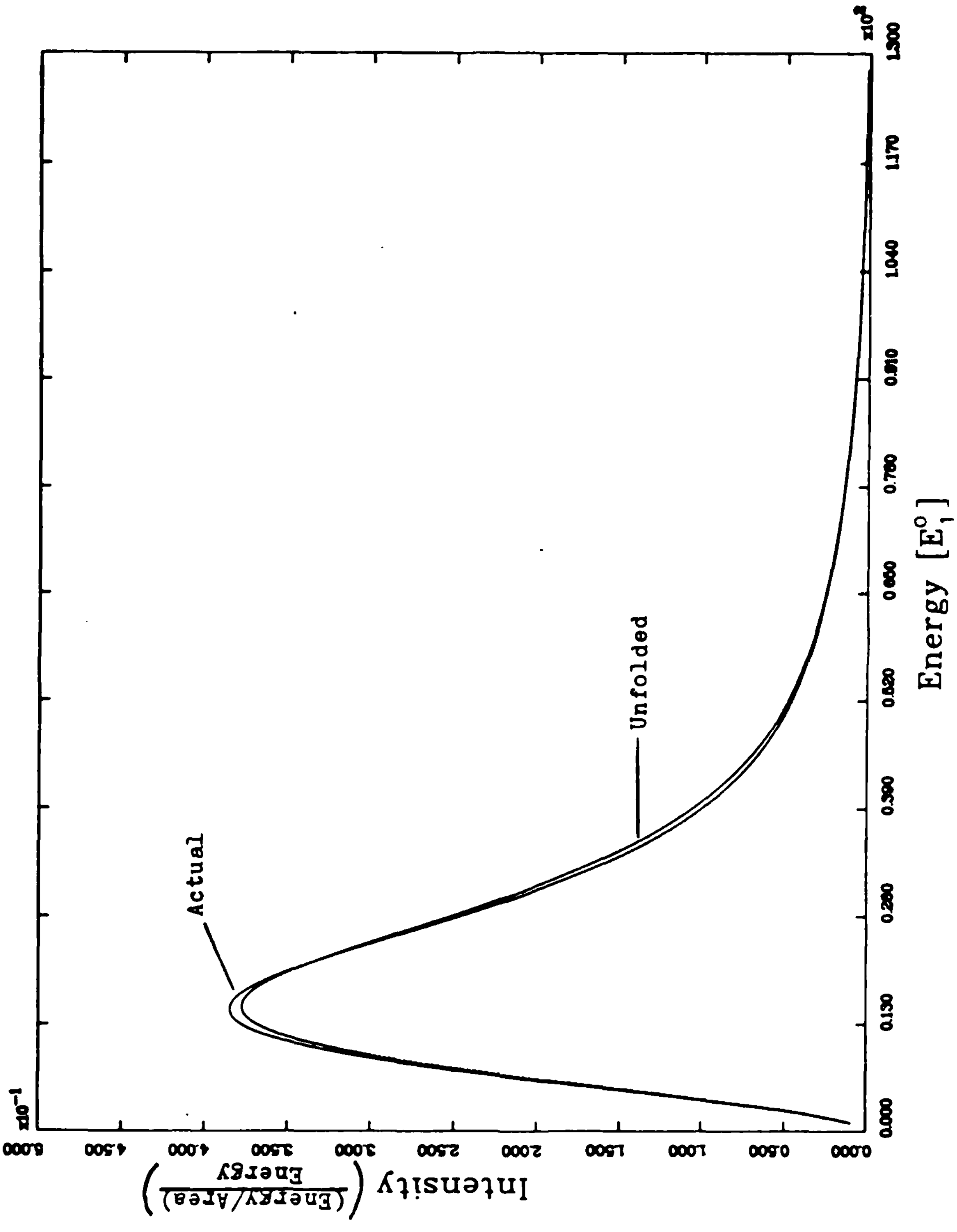


Figure 24. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP03

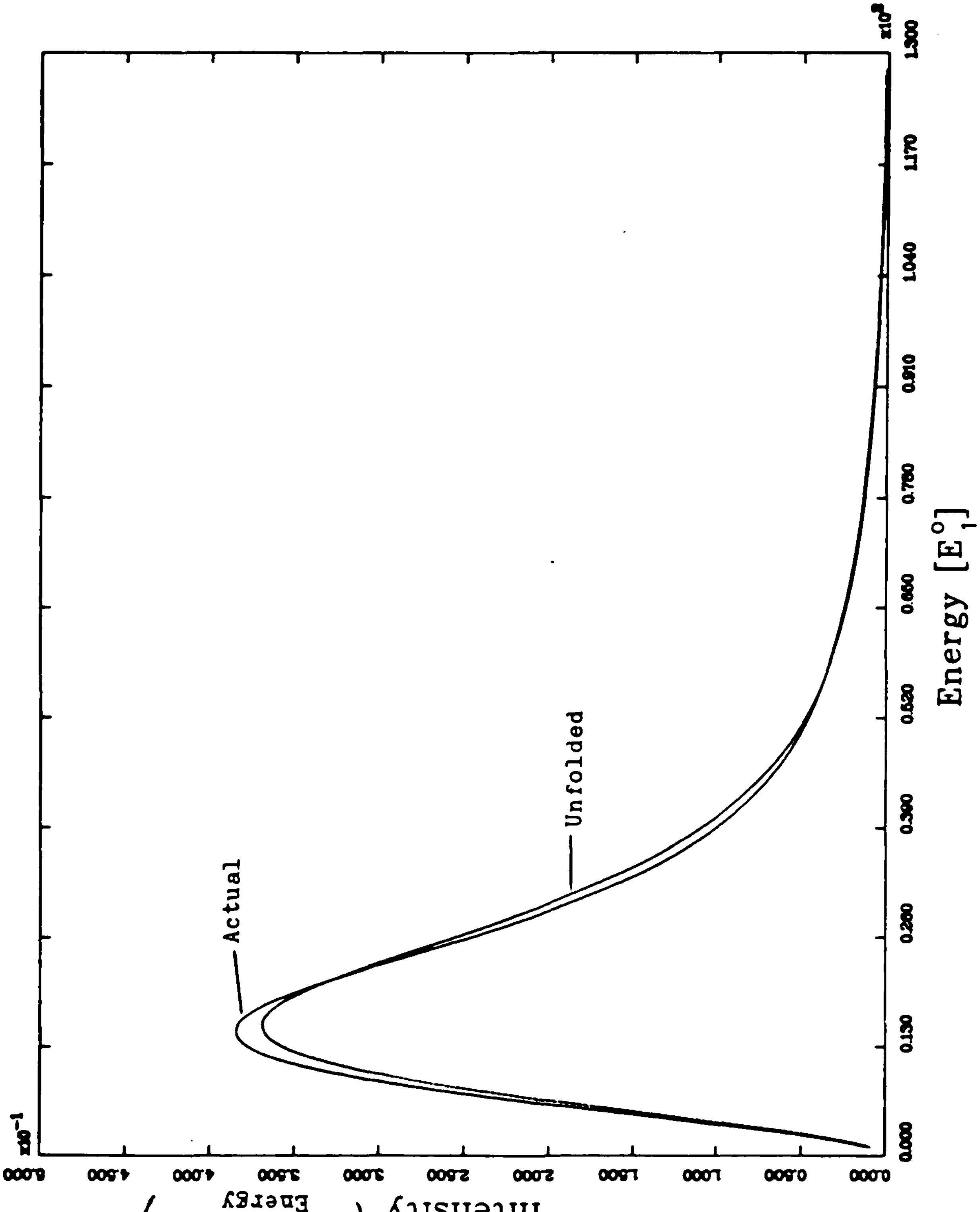


Figure 25. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP04

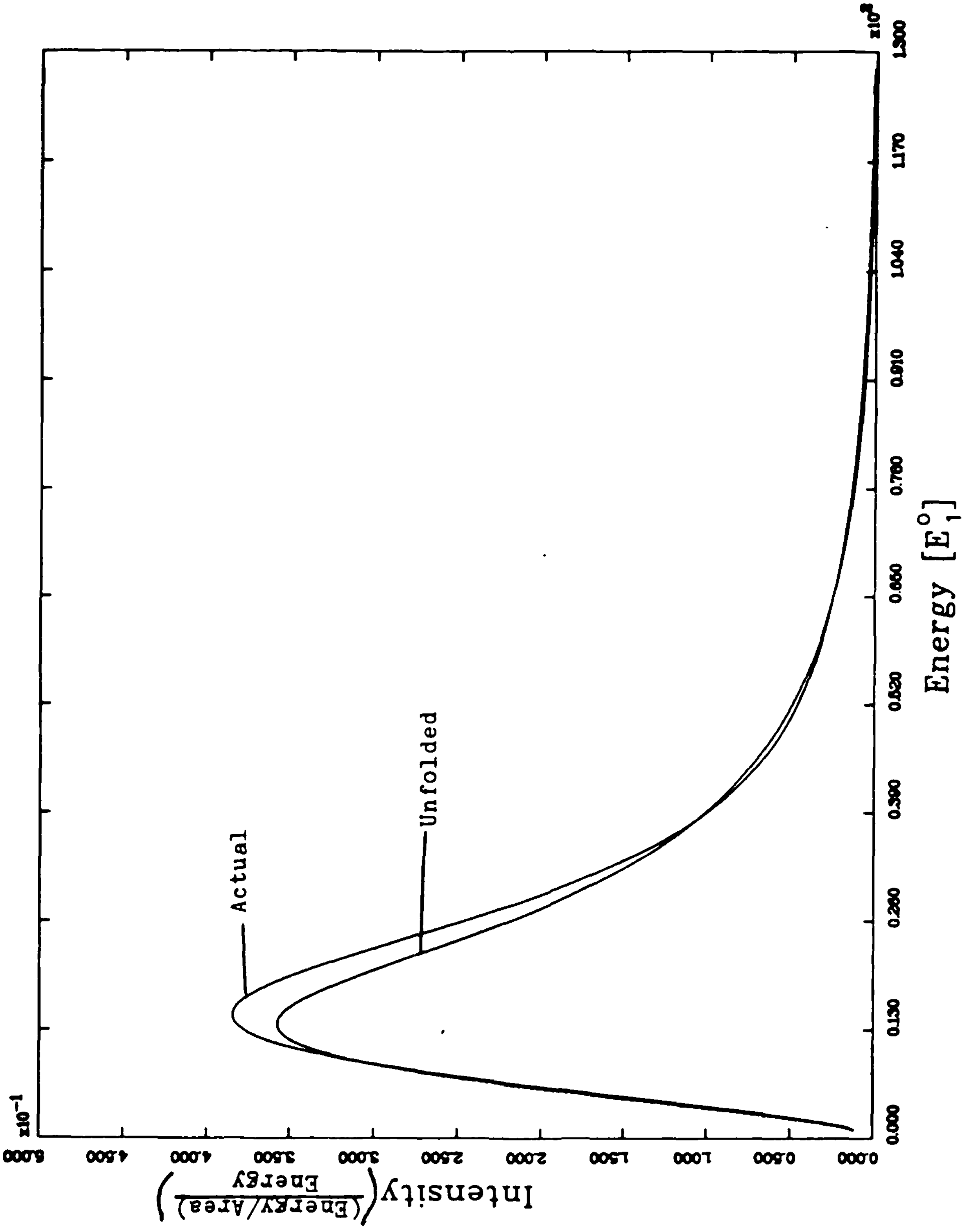


Figure 26. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP05

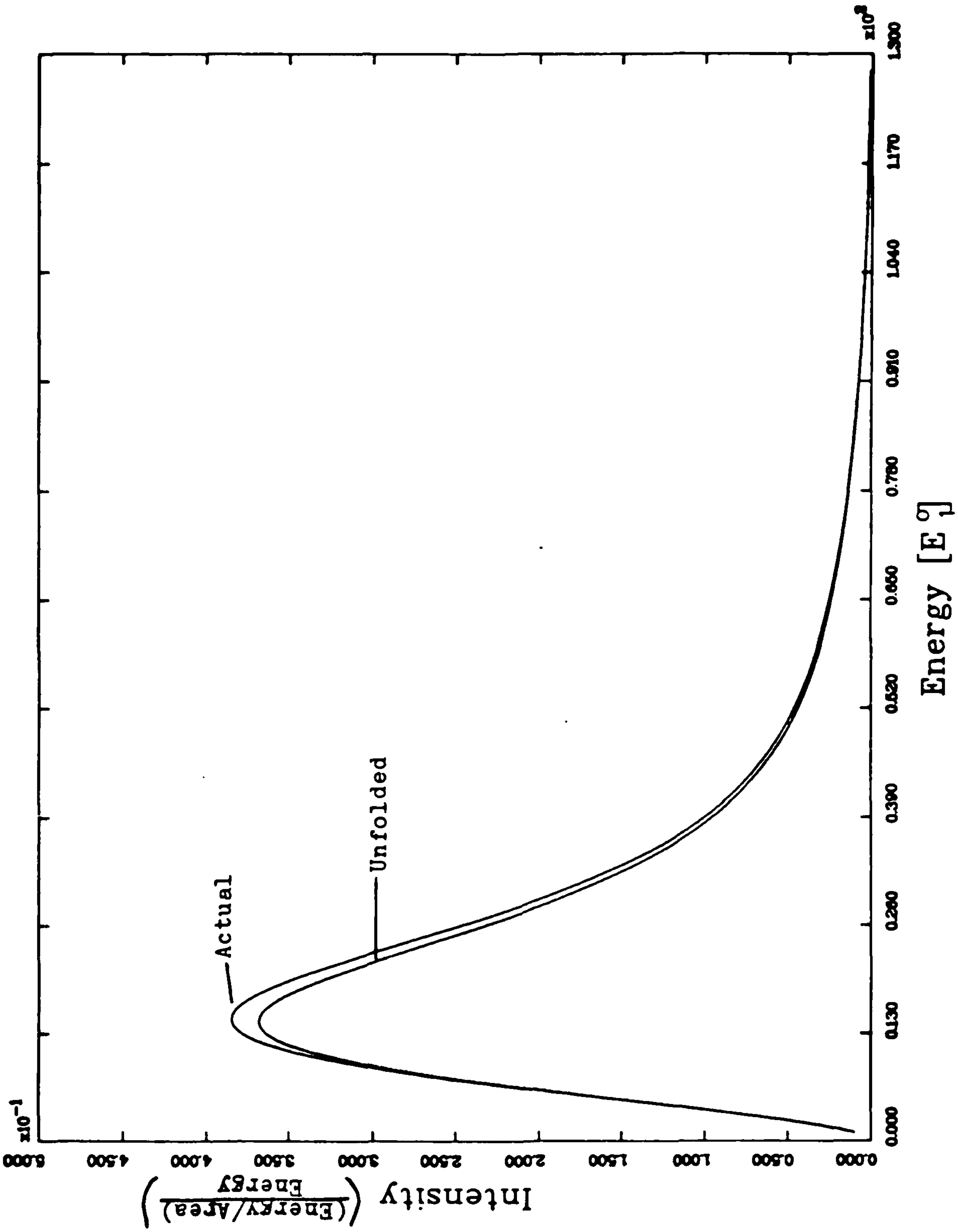


Figure 27. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP06

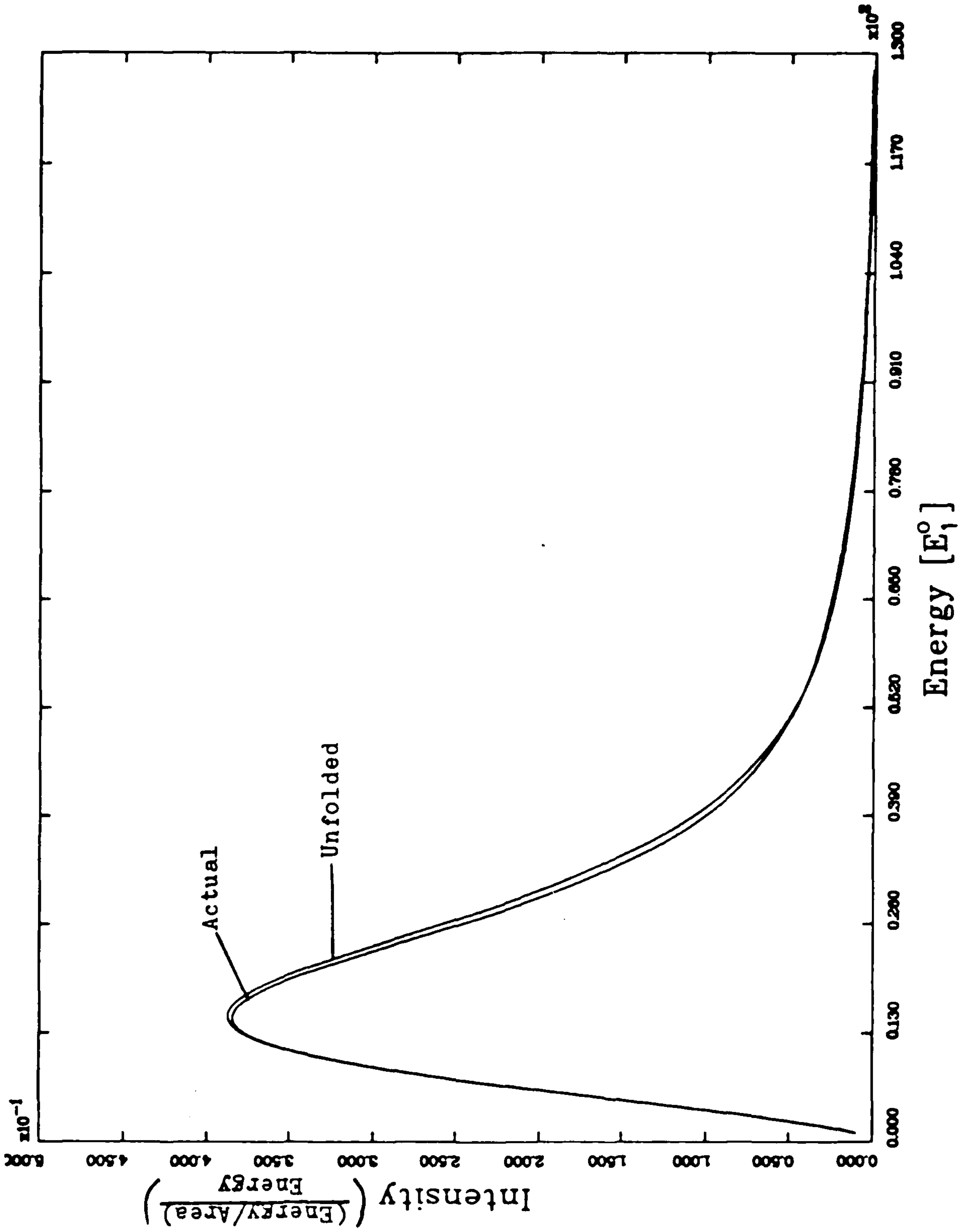


Figure 28. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP07

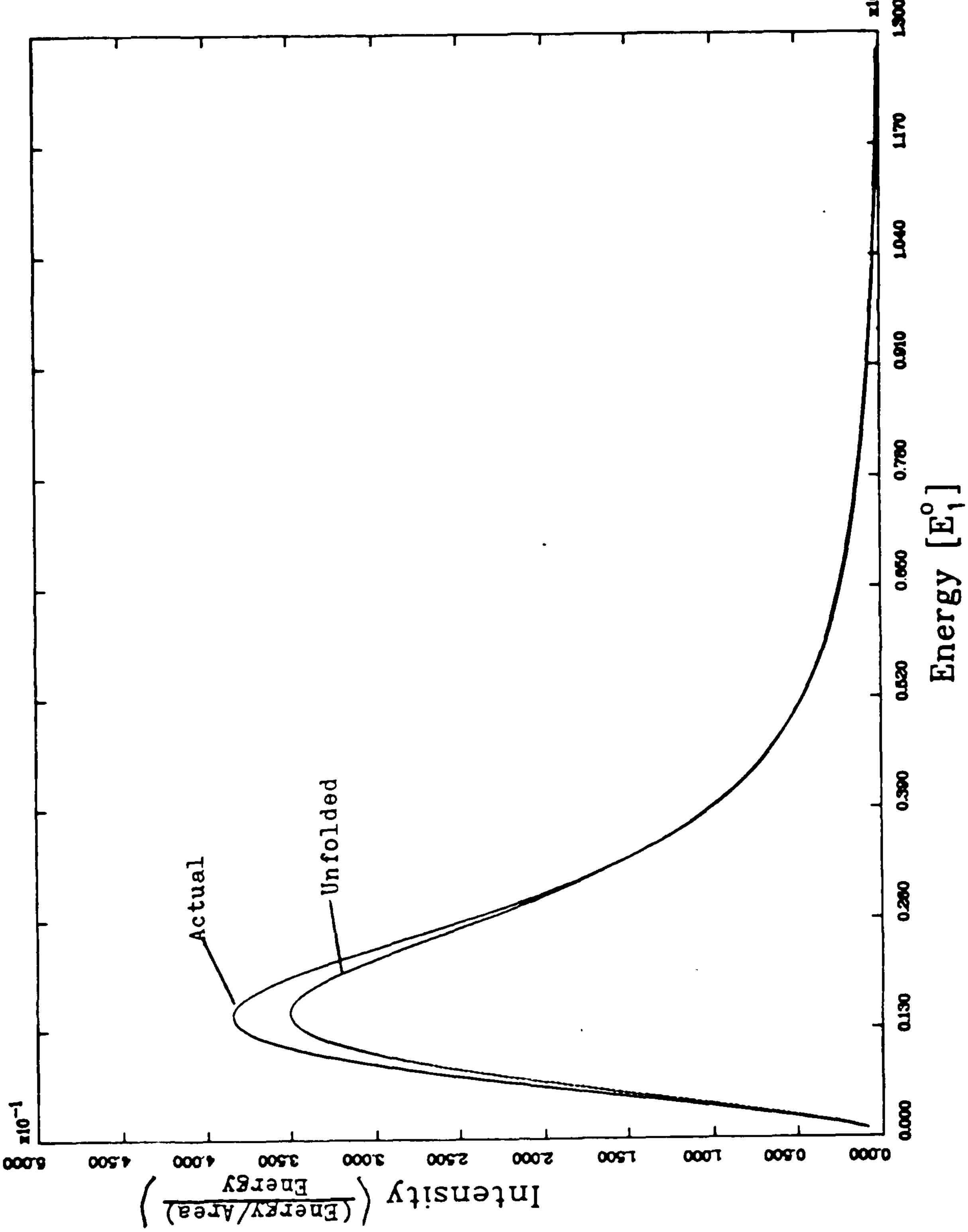


Figure 29. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP08

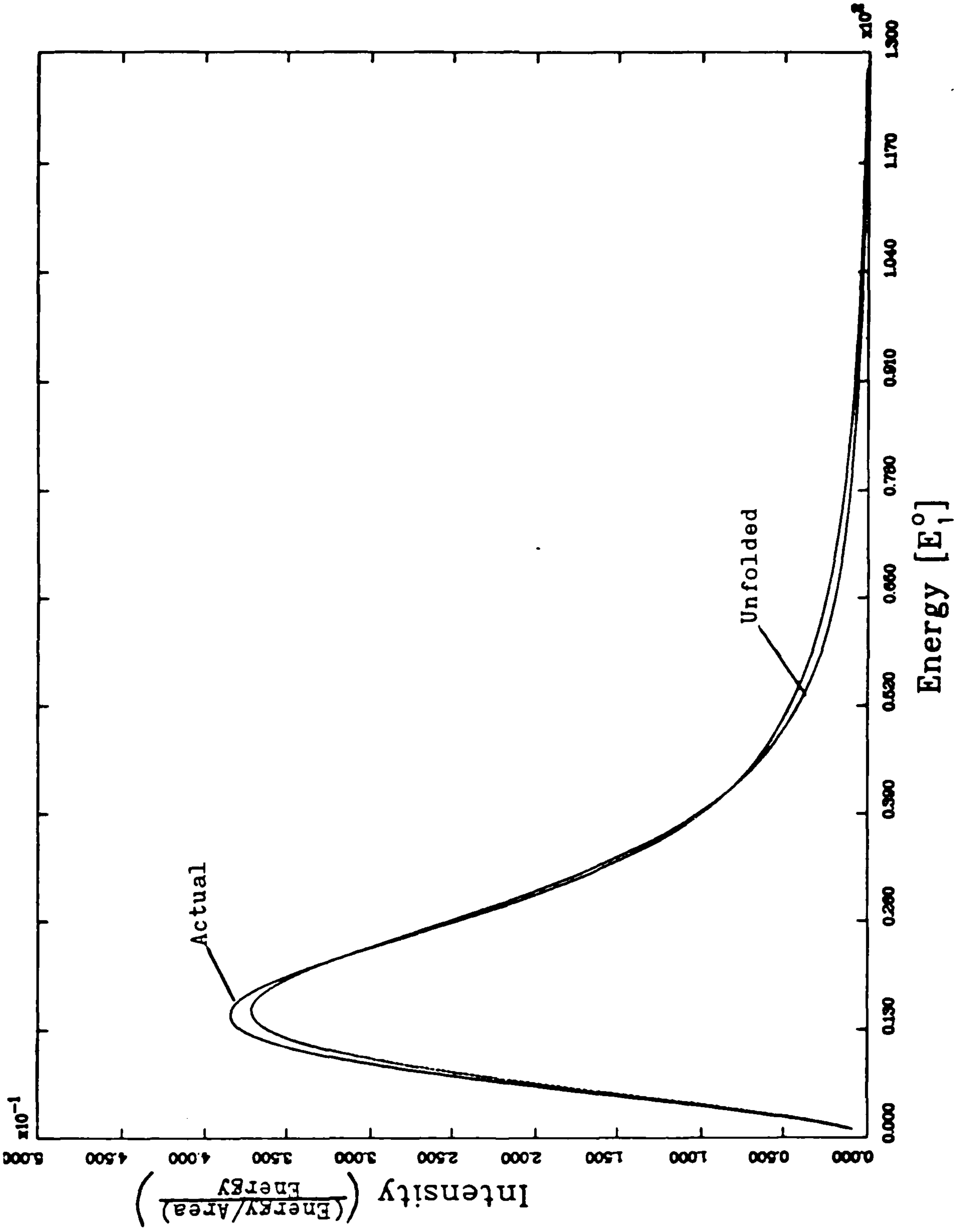


Figure 30. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP09

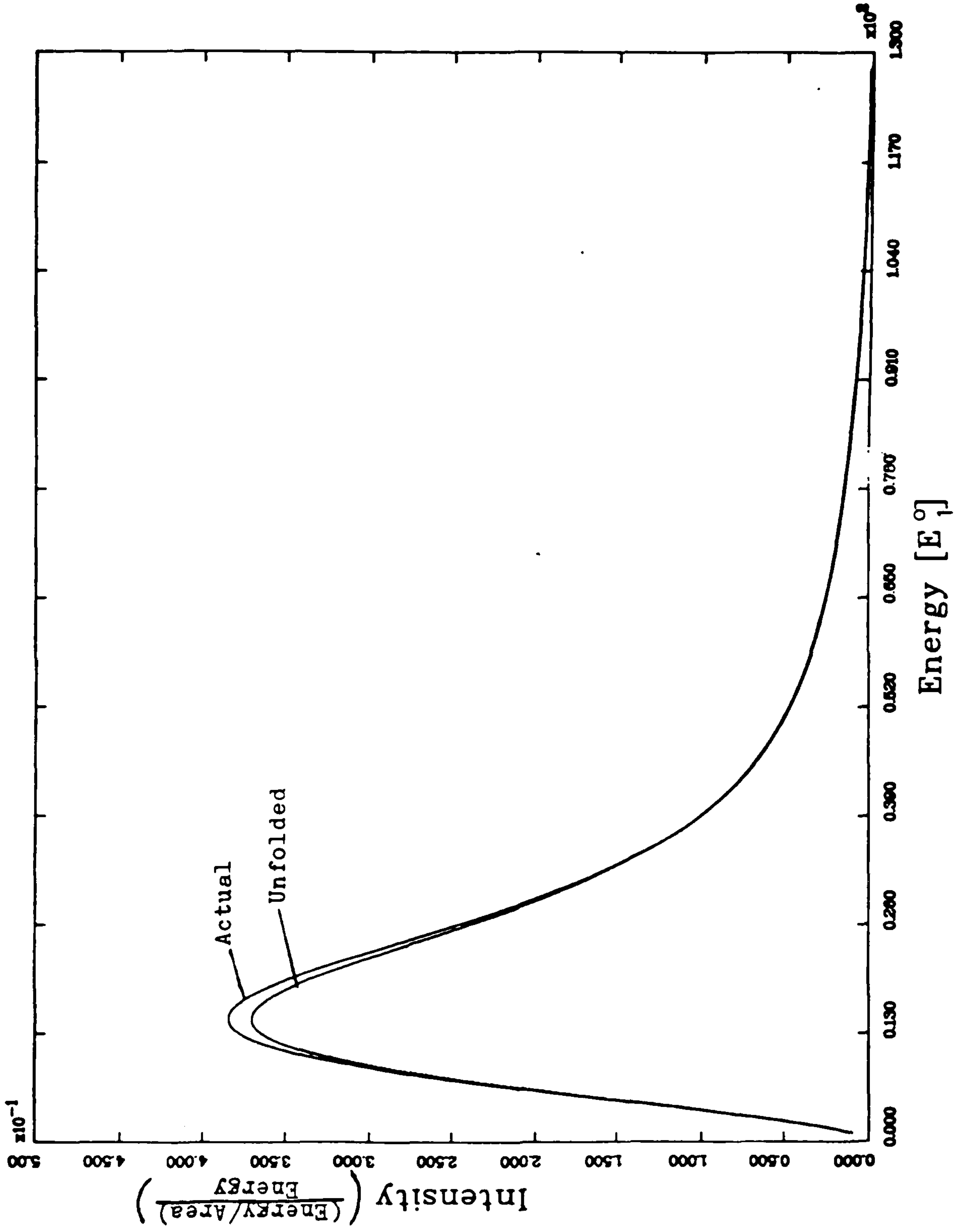


Figure 31. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NP010

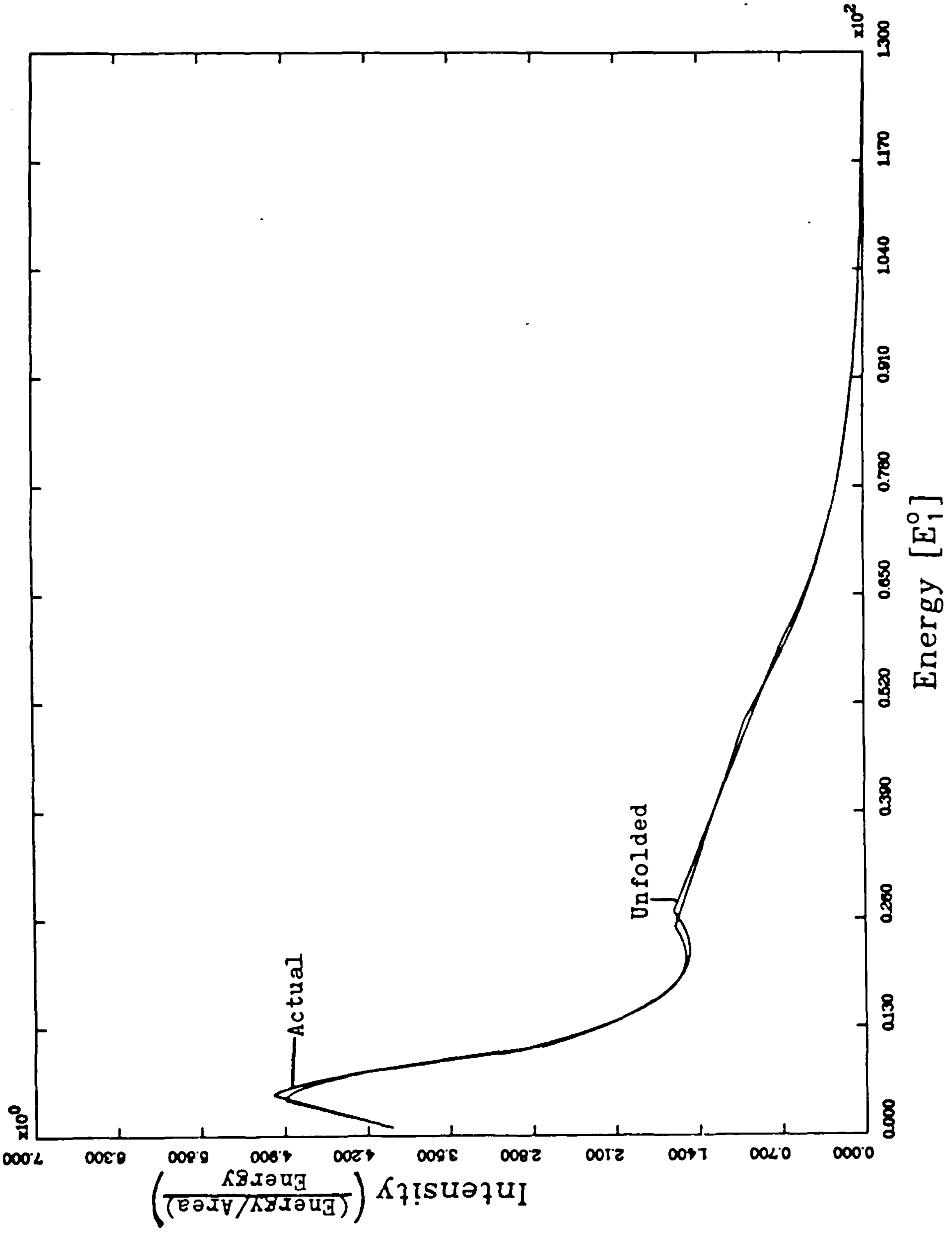


Figure 32. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC0

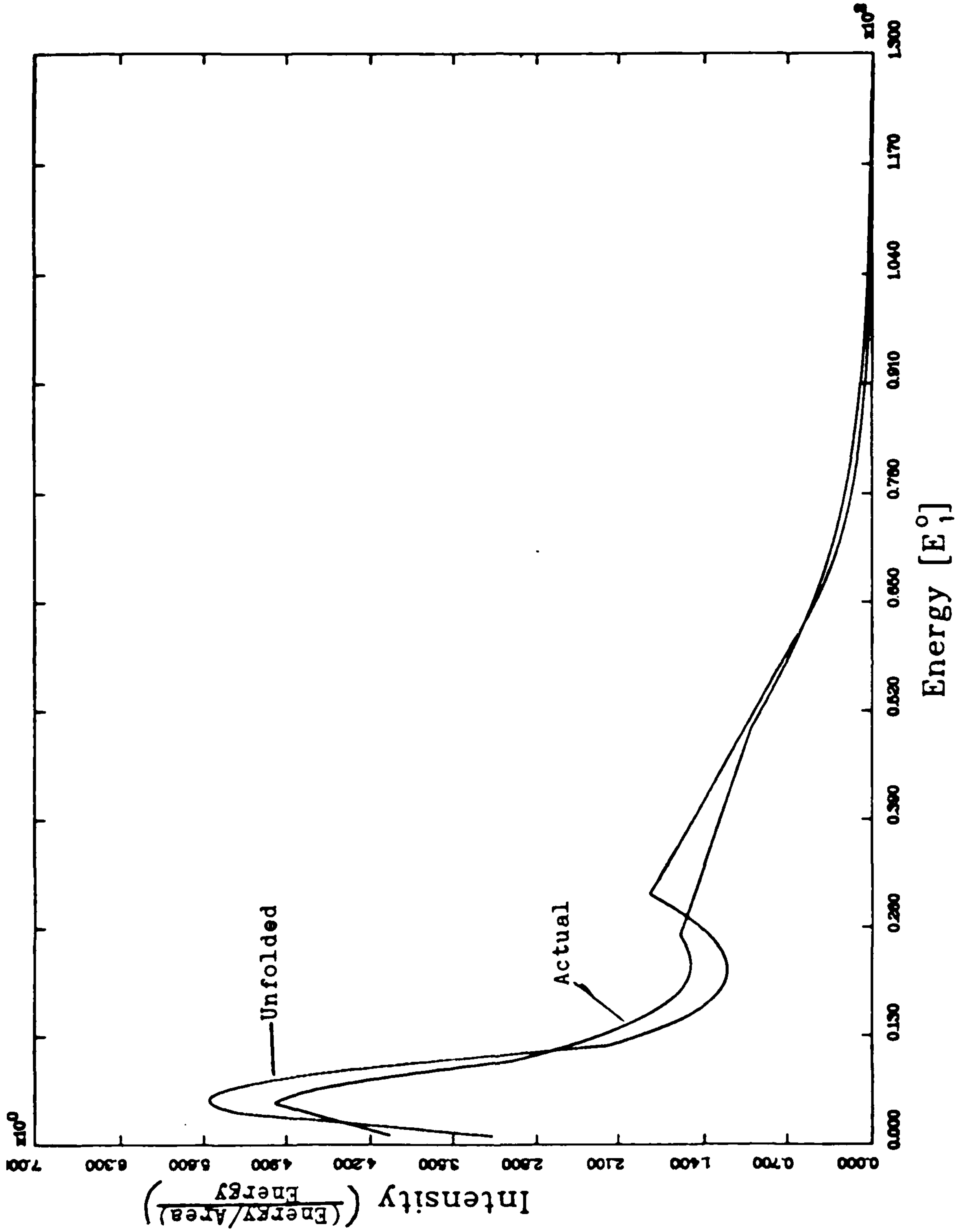


Figure 33. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC01

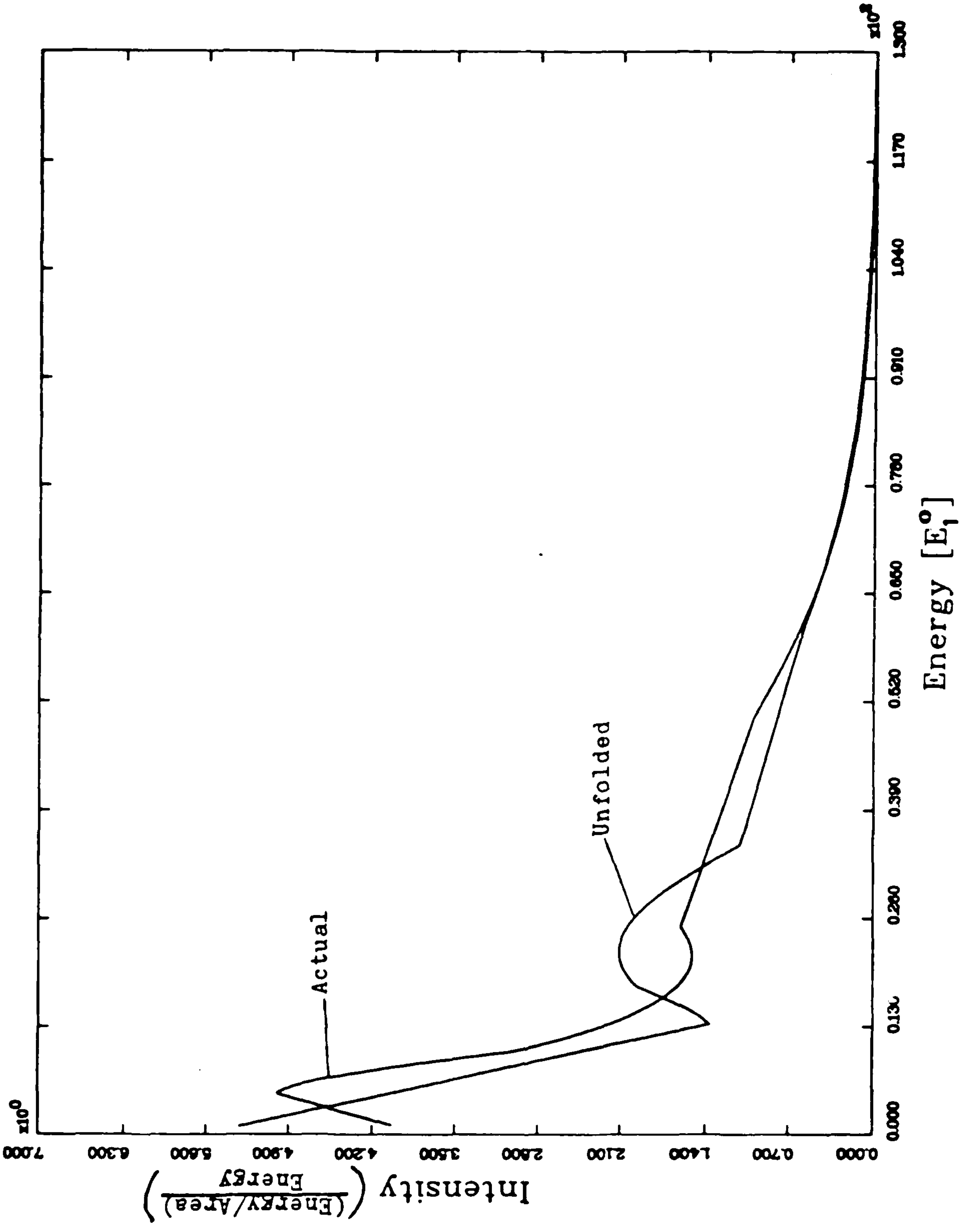


Figure 34. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC02

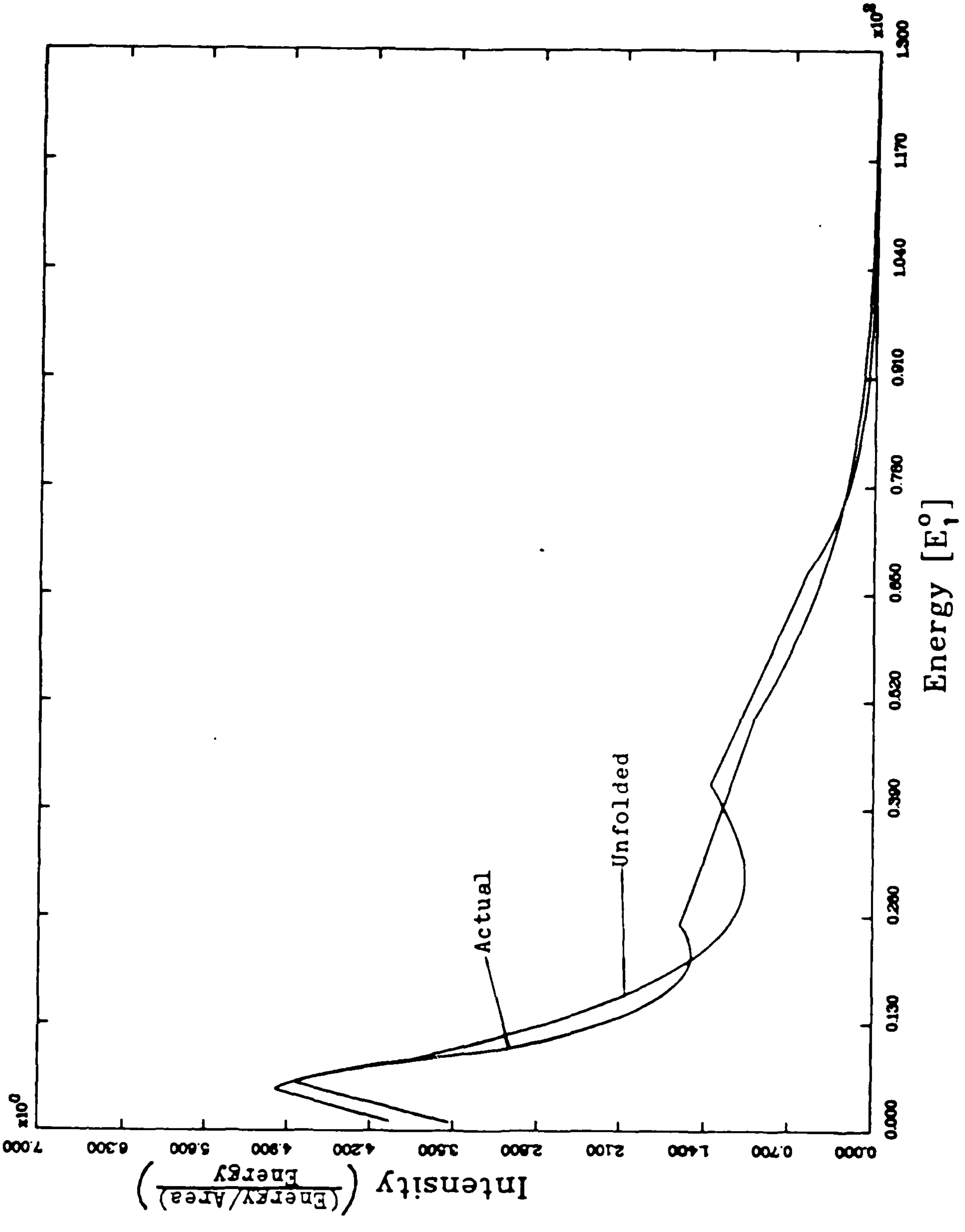


Figure 35. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC03

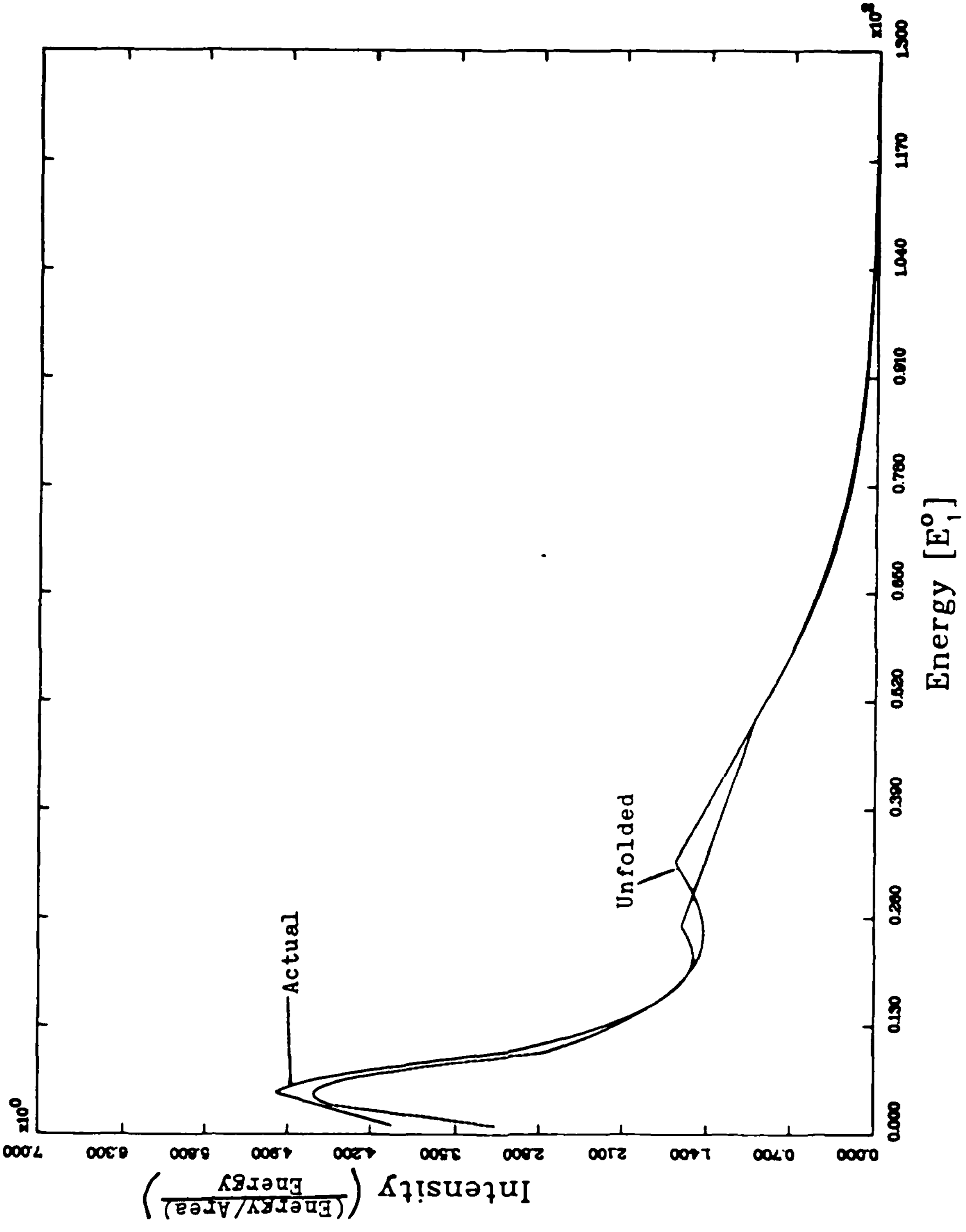


Figure 36. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC04

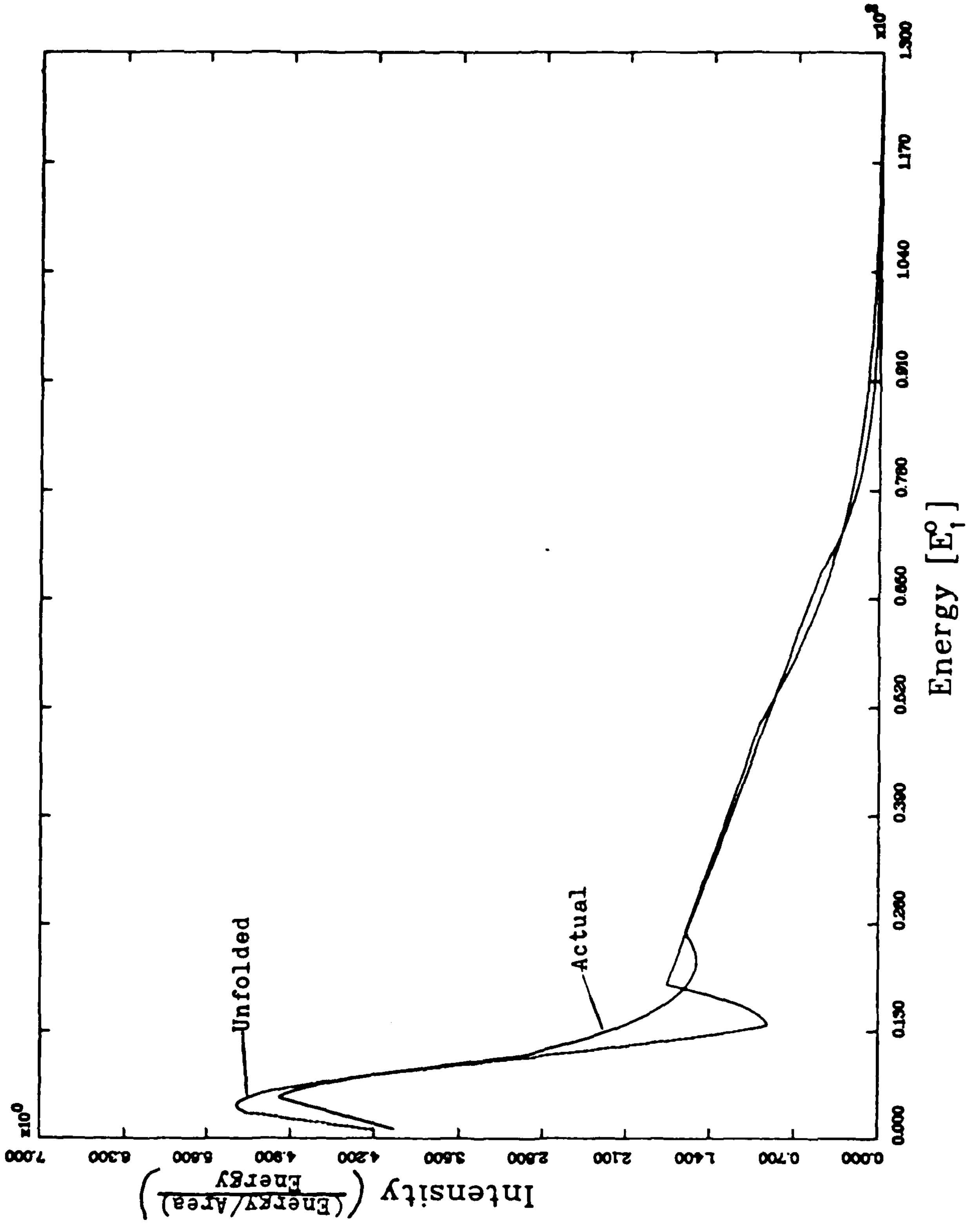


Figure 37. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC05

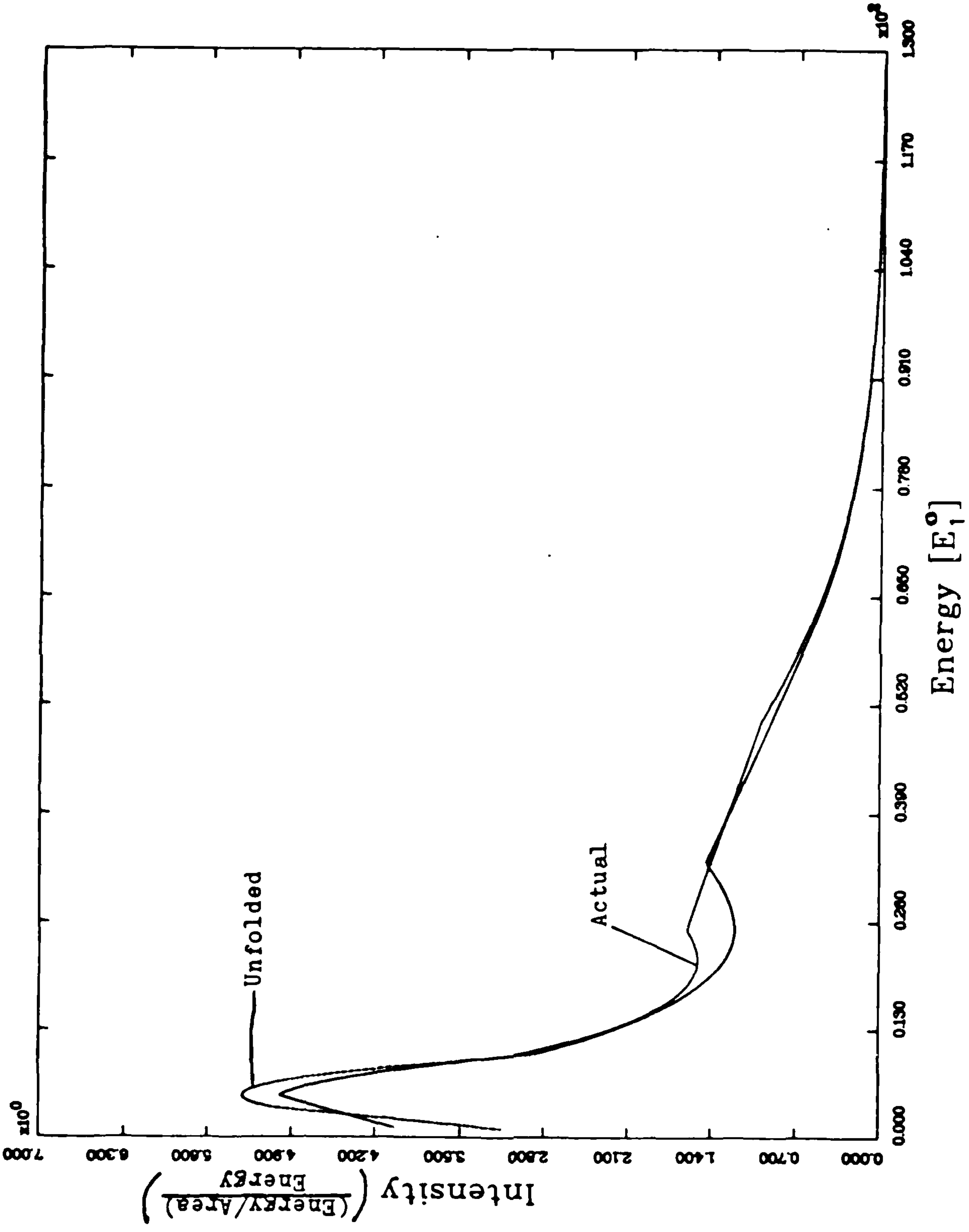


Figure 38. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC06

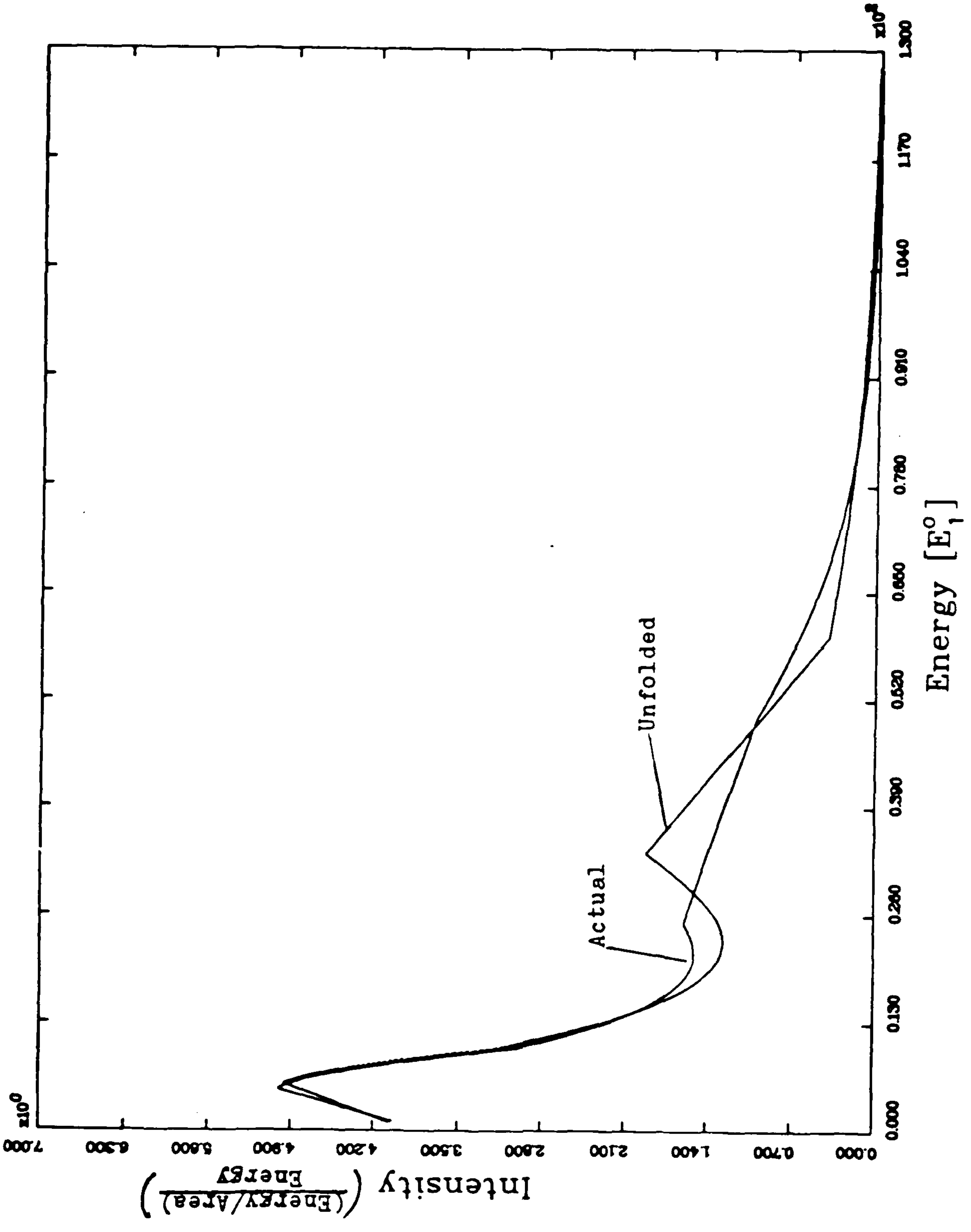


Figure 39. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC07

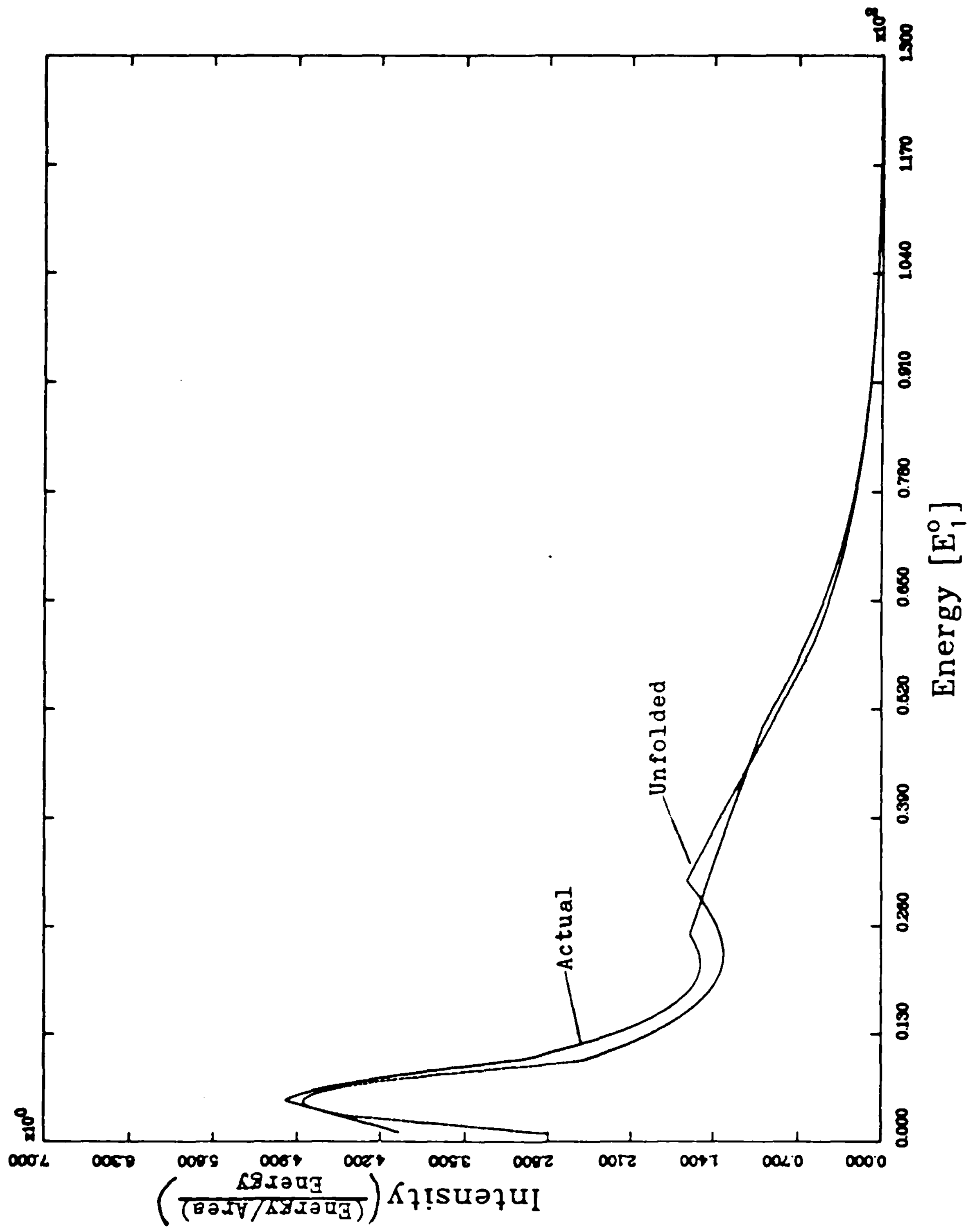


Figure 40. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC08

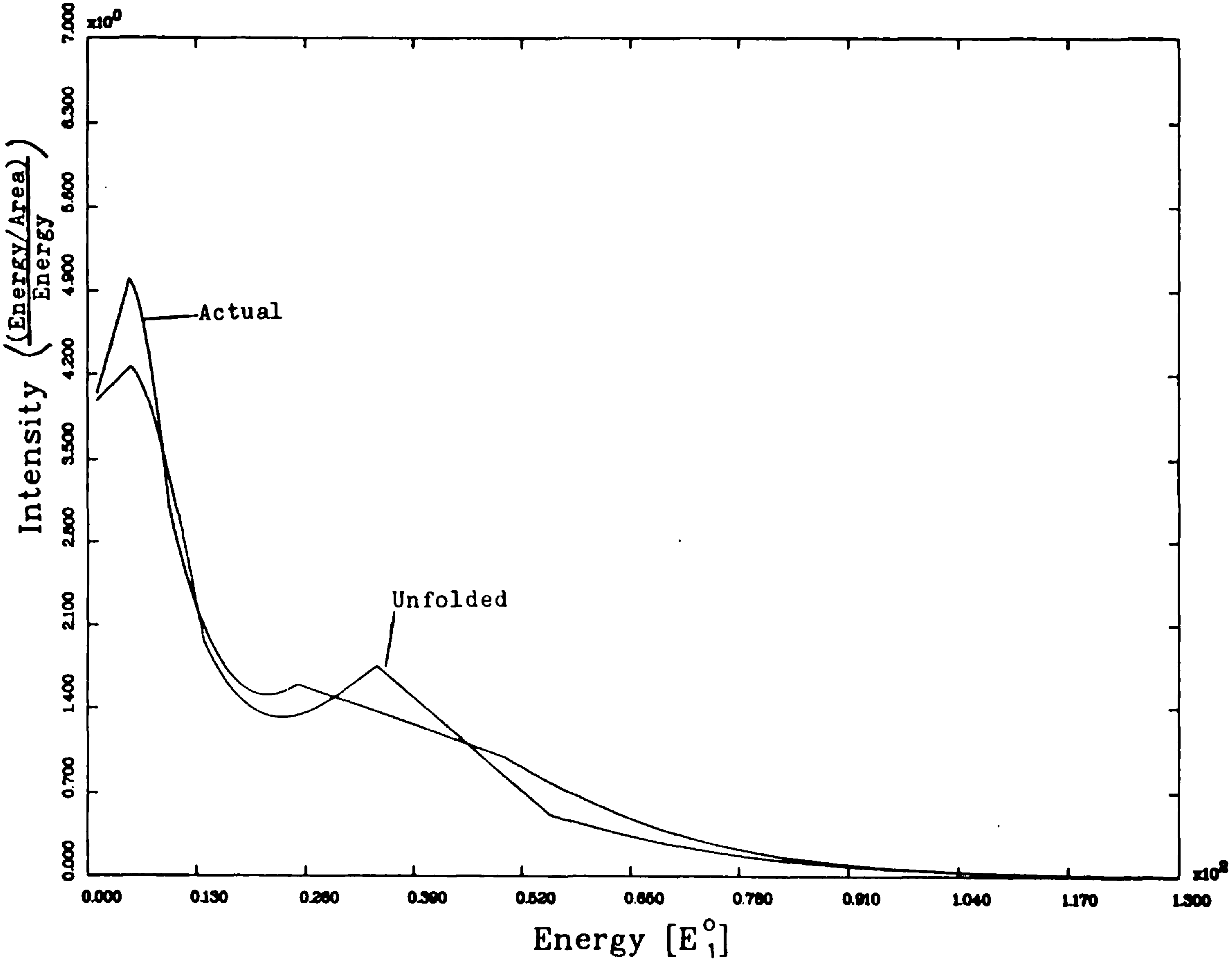


Figure 41. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC09

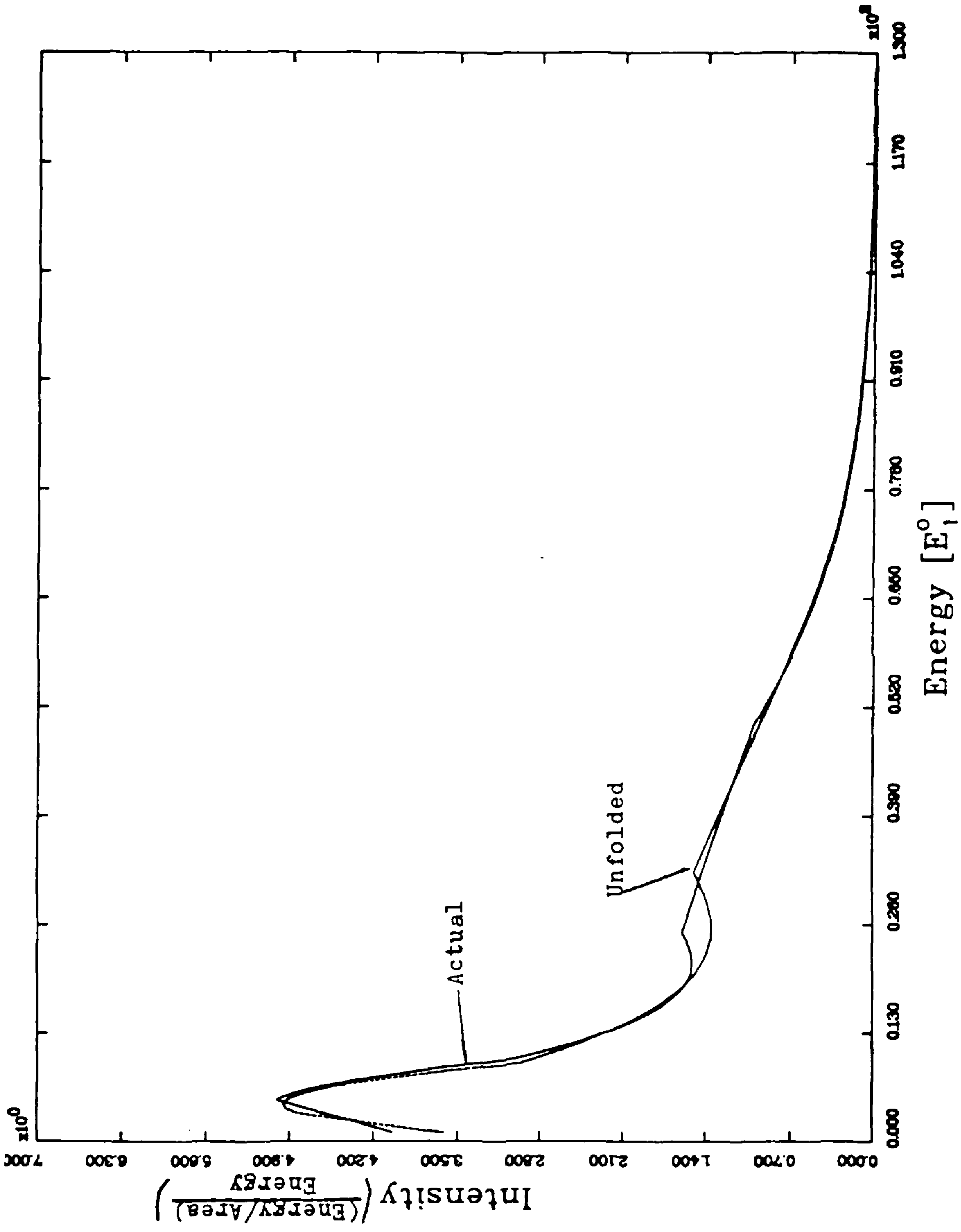


Figure 42. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case NC10

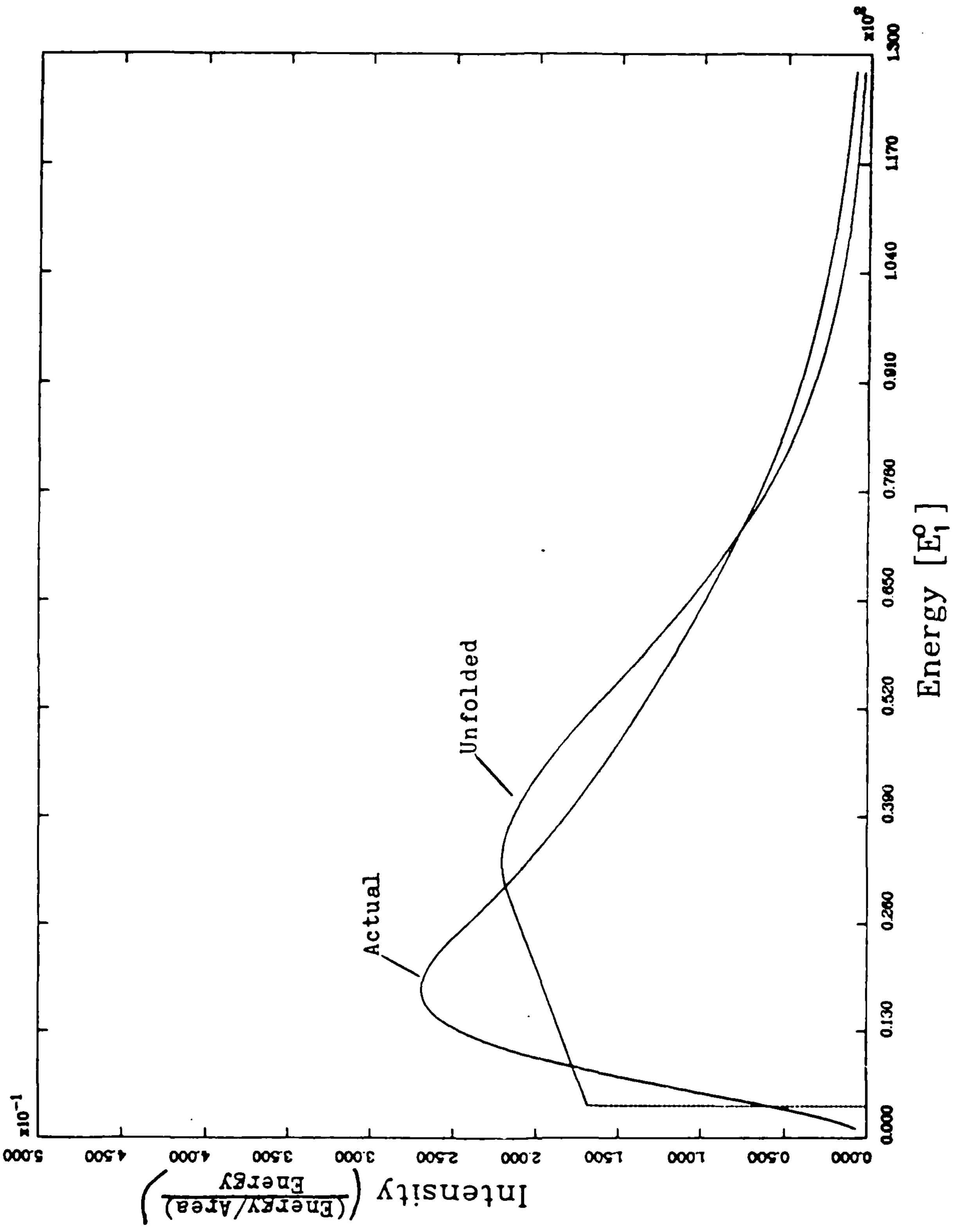


Figure 43. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case PC1

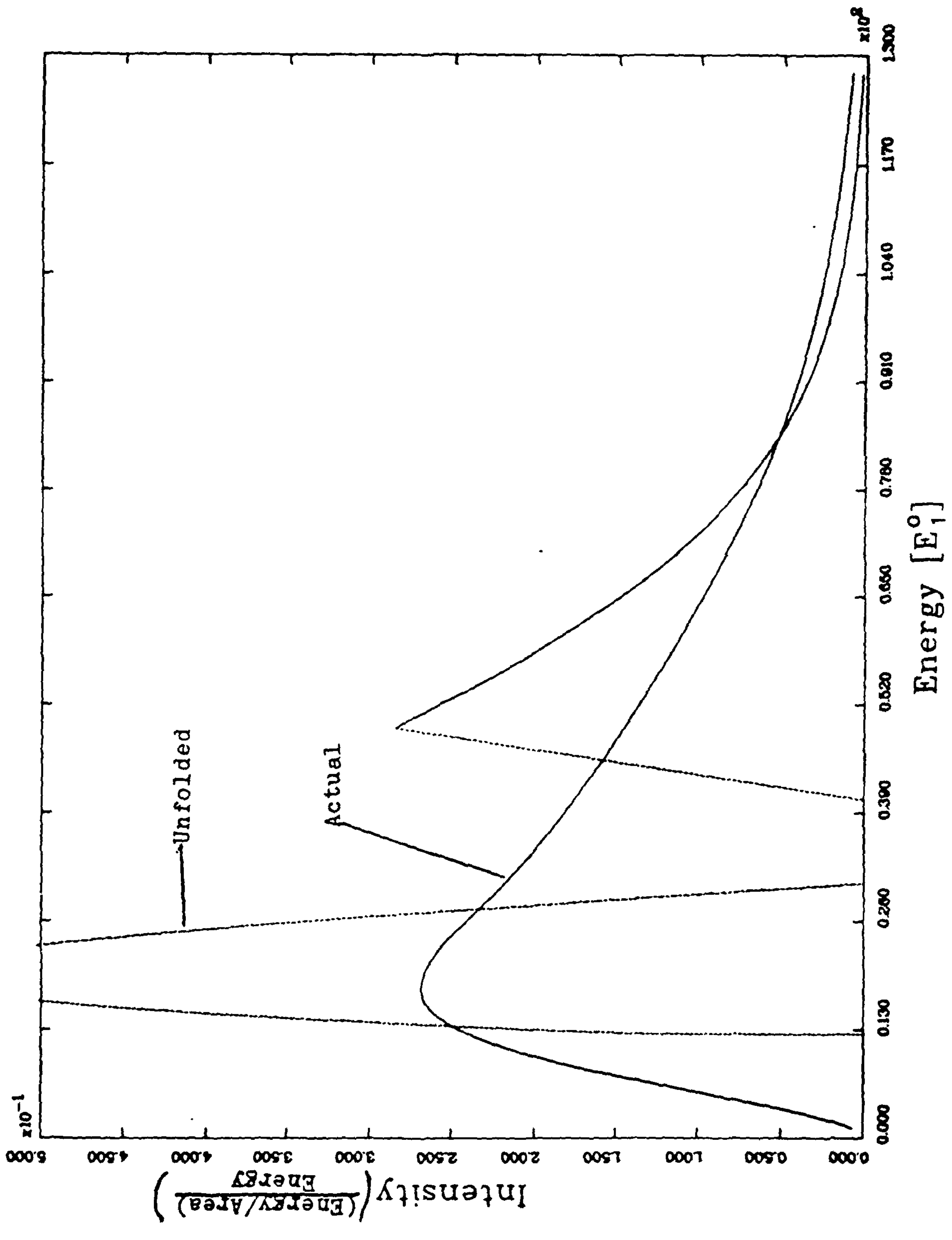


Figure 44. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case PC2

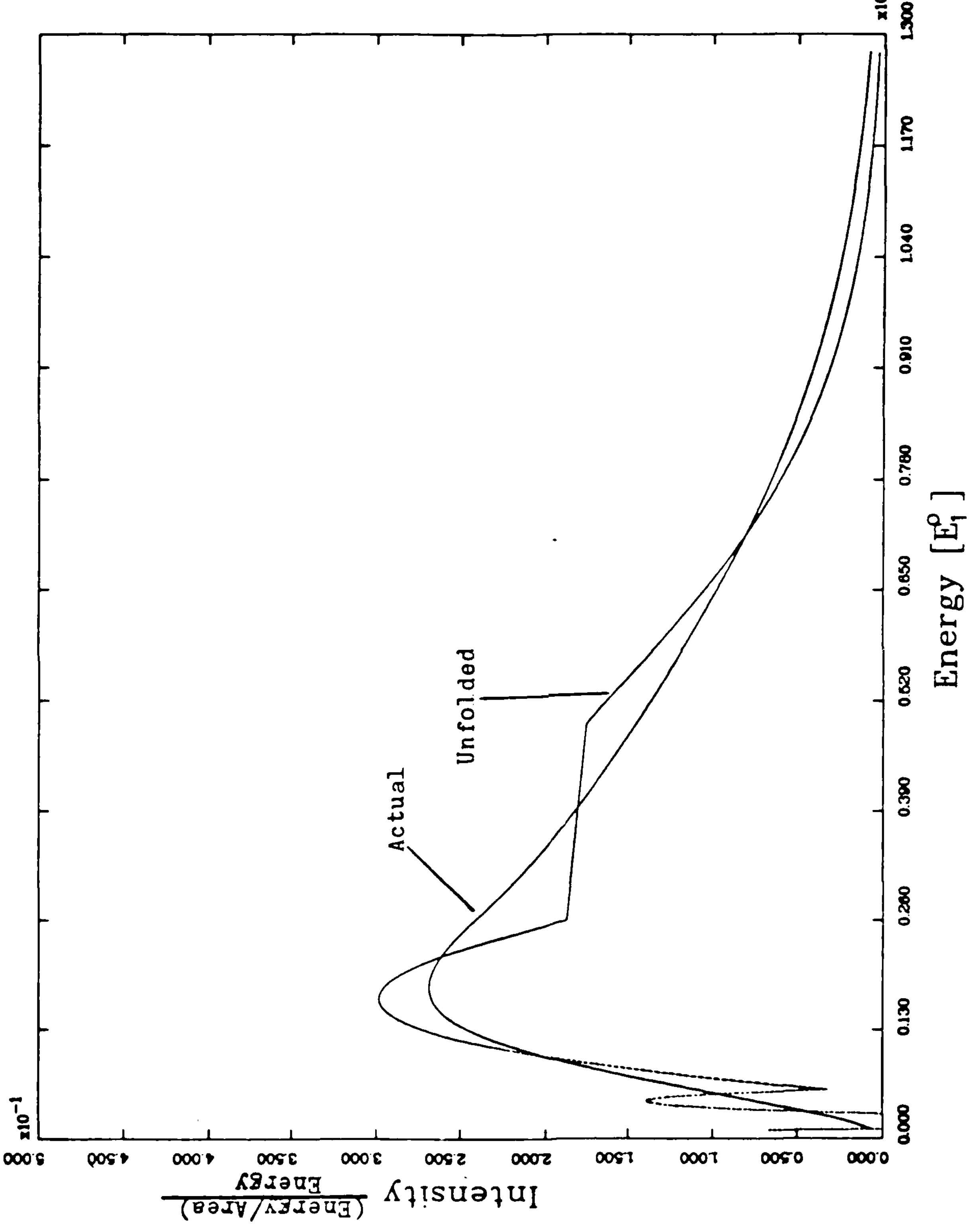


Figure 45. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case PC3

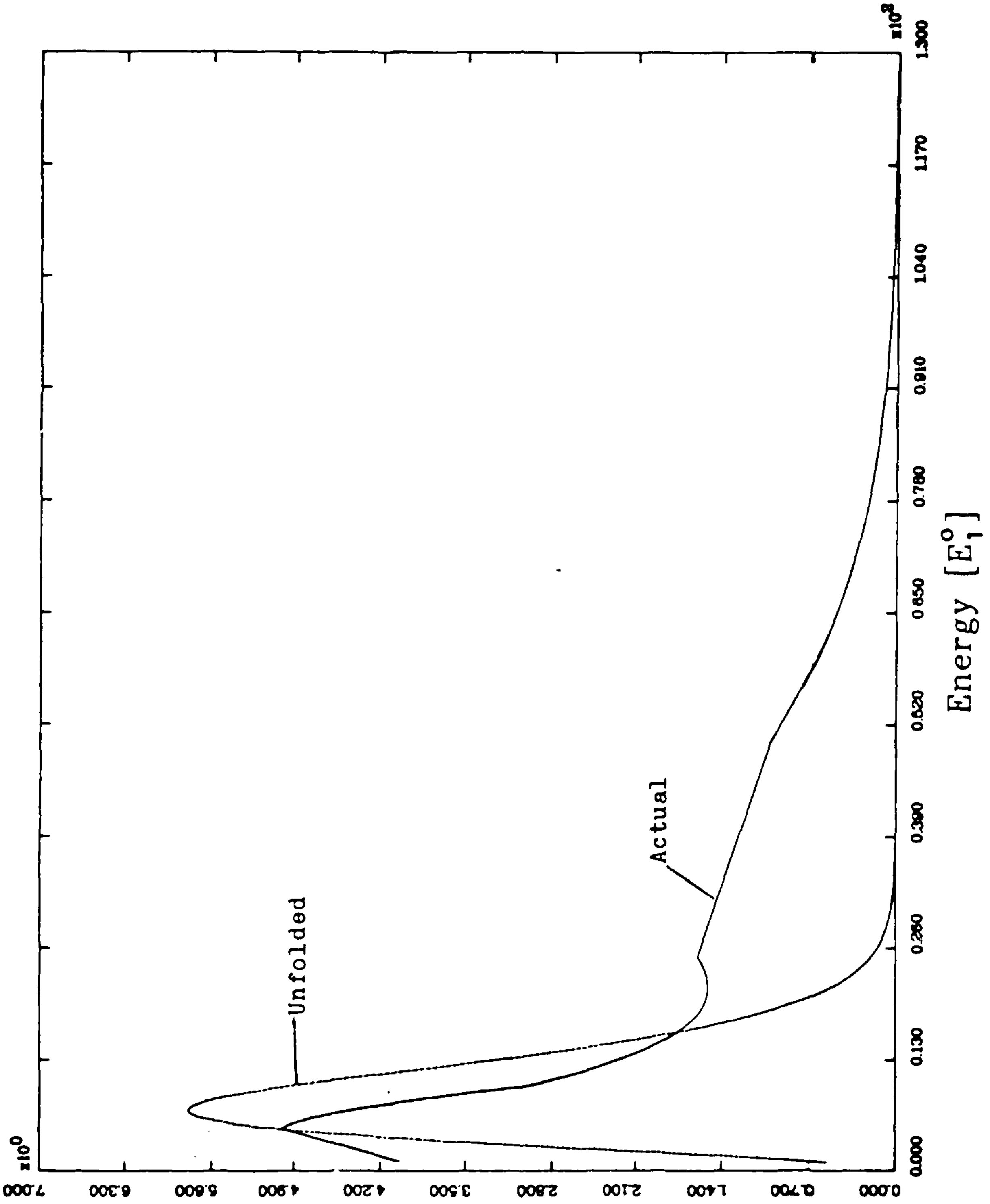


Figure 46. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case CP1

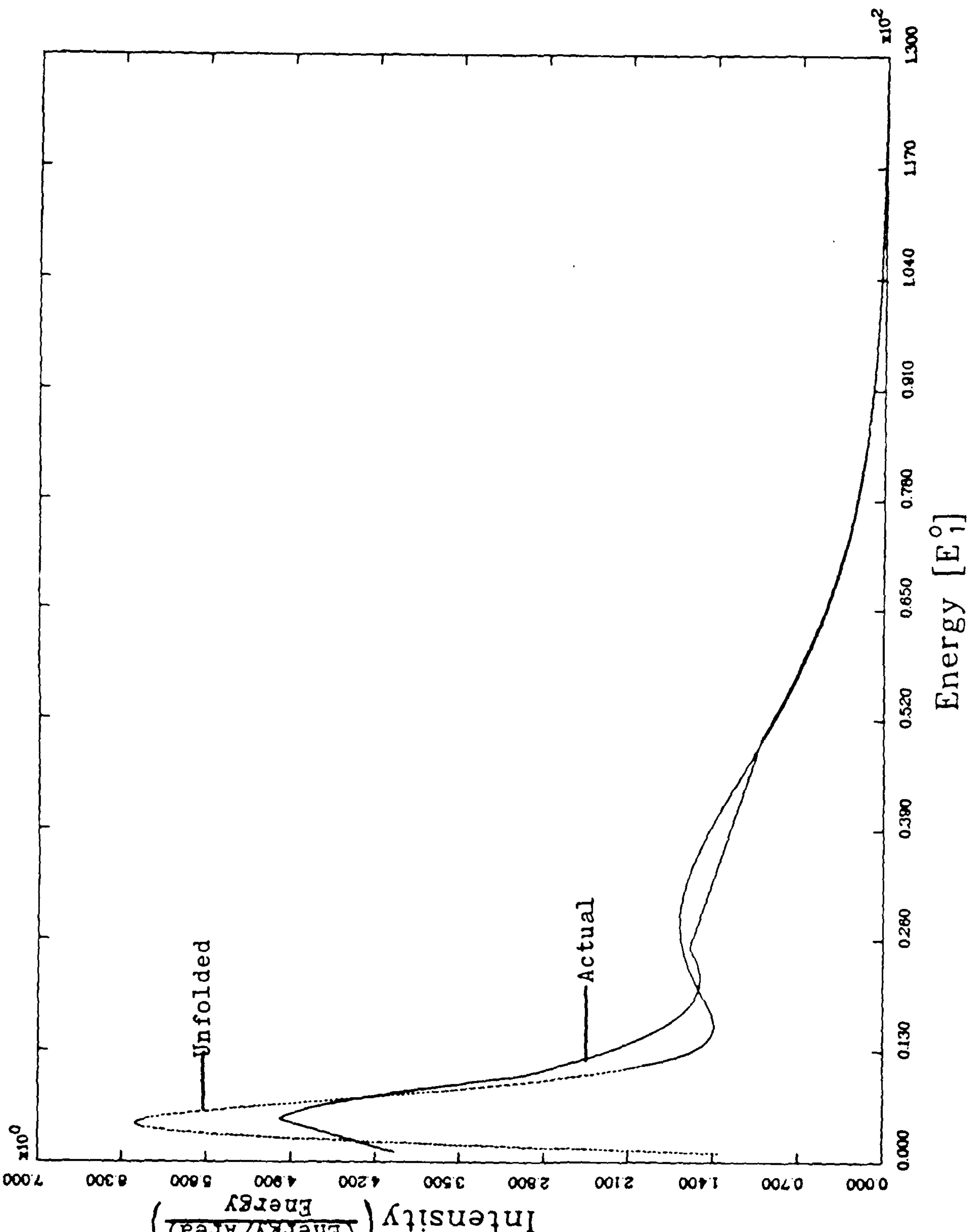


Figure 47. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case CP2

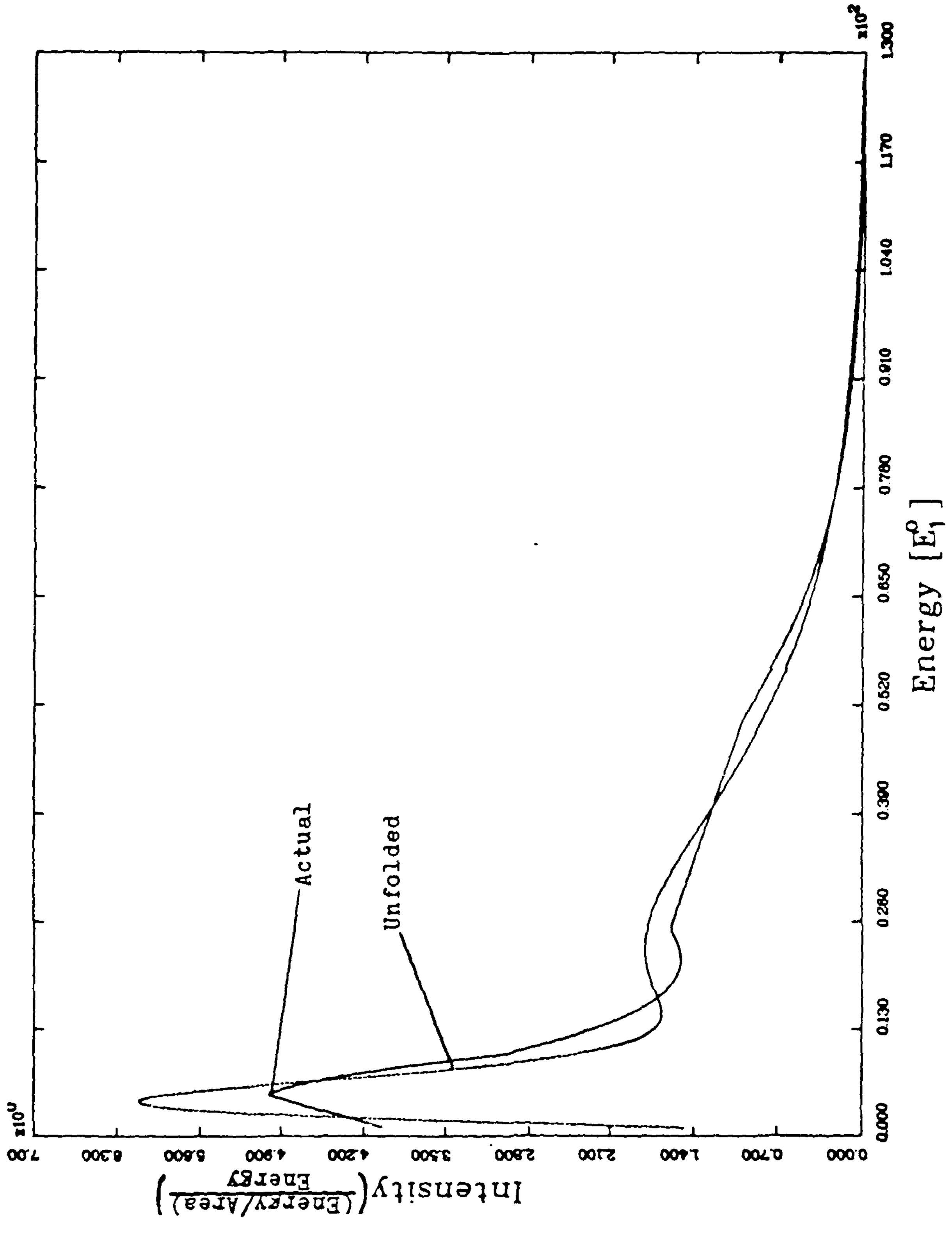


Figure 48. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case CP3

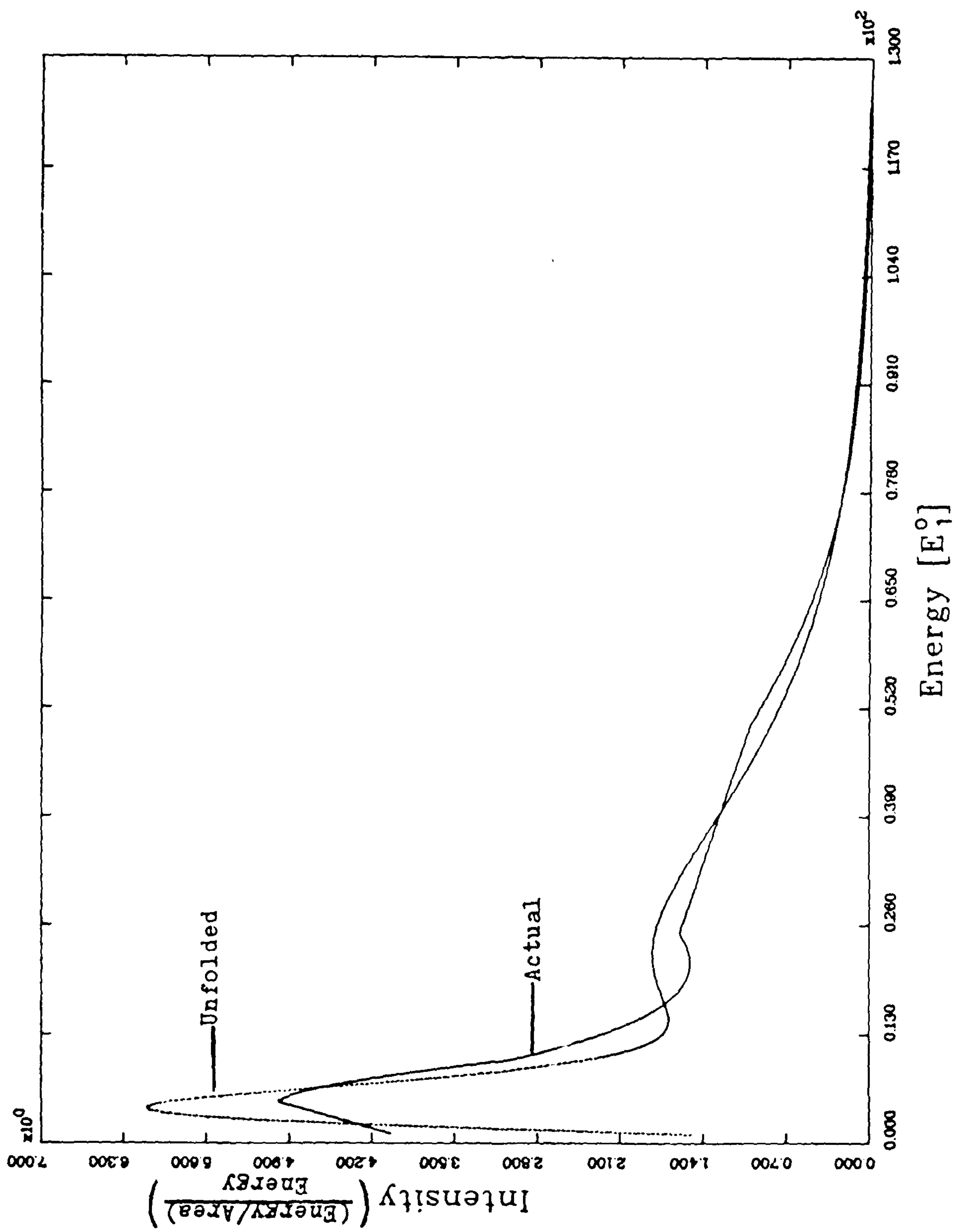


Figure 49. Linear Plot of the Actual and Unfolded Spectra Versus Energy for Case CP4

Appendix G: Figure of Merit Study

In order to determine the accuracy of the approximation, eight figures of merit were considered. The goal of this study was to determine if any of the figures of merit could be related to χ^2. By relating the figure of merit to χ^2, a method for determining the accuracy of the unfolding procedure may be established. The eight figures of merit are as follows: the average absolute error, AAE; the average absolute relative error, AARE; the root mean square average absolute error, RMSAAE; the root mean square average relative error, RMSARE; the weighted average absolute error, WAAE; the weighted average absolute relative error, WAARE; the weighted root mean square average absolute error, WRMSAAE; and the weighted root mean square average relative error, WRMSARE. Equations (22) thru (29) present the figures of merit studied and the approximations used by LCDR. Kirk Mathews to produce a computer program to calculate the figures of merit. The program was validated using case BC1 and the TK Solver mathematics package.

$$AE \begin{cases} = \int \left| S_u(E) - S_a(E) \right| dE \\ \approx \sum_k \left| S_u(E_k) - S_a(E_k) \right| \end{cases} \tag{22}$$

$$RE \begin{cases} = \int \left\{ \dfrac{\left| S_u(E) - S_a(E) \right|}{\left[\dfrac{S_u(E) + S_a(E)}{2} \right]} \right\} dE \\ \approx \sum_k \left\{ \dfrac{\left| S_u(E_k) - S_a(E_k) \right|}{\left[\dfrac{S_u(E_k) + S_a(E_k)}{2} \right]} \right\} \end{cases} \tag{23}$$

Due to the fact all the energy bin widths were equally spaced, the average absolute error and average relative error can be calculated by dividing Equations (22) and (23) by nk, the number of energy bins used to evaluate the integrals.

Thus

$$AAE = \frac{AE}{nk} \tag{24}$$

$$AARE = \frac{RE}{nk} \tag{25}$$

$$RMSAAE = \left\{\frac{AE^2}{nk}\right\}^{1/2} \tag{26}$$

$$RMSARE = \left\{\frac{RE^2}{nk}\right\}^{1/2} \tag{27}$$

Finally the weighted figures of merit were calculated by weighting the absolute error and relative error with a factor of 1/E.

Thus

$$WAAE \left\{ \begin{array}{l} = \int \left\{ \dfrac{|S_u(E)-S_a(E)|}{E} \right\} dE \\ \approx \sum\limits_k \left\{ \dfrac{|S_u(E_k)-S_a(E_k)|}{E_k} \right\} \end{array} \right. \tag{28}$$

$$WARE \left\{ \begin{array}{l} = \int \left\{ \dfrac{|S_u(E)-S_a(E)|}{\left[\dfrac{S_u(E)+S_a(E)}{2}\right]} \right\} (\dfrac{1}{E}) dE \\ \approx \sum\limits_k \left\{ \dfrac{|S_u(E_k)-S_a(E_k)|}{\left[\dfrac{S_u(E_k)+S_a(E_k)}{2}\right]} \right\} (\dfrac{1}{E_k}) \end{array} \right. \tag{29}$$

The weighted figures of merit can then be calculated using Equations (24) thru (27) by replacing the absolute error and the relative error with the weighted absolute error and the weighted relative error respectively.

The results of this study are presented in Table XIV.

TABLE XIV

Comparison Of Figures Of Merit For Cases Studied (a)

Cs	X^2	AAE	ARE	RMS AE	RMS RE	WAAE	WAARE	WRMS AE	WRMS RE
NP0	0.010	4e-4	0.012	7e-4	0.017	5e-4	0.010	9e-4	0.010
NC0	0.020	0.017	0.047	0.027	0.066	0.028	0.020	0.042	0.035
BC4	0.026	0.001	0.001	0.003	0.002	0.005	0.002	0.008	0.002
BC2	5.4	0.014	0.013	0.029	0.017	0.037	0.014	0.053	0.018
NC04	5.5	0.06	0.11	0.12	0.13	0.23	0.091	0.34	0.11
NP04	6.0	0.005	0.095	0.008	0.11	0.009	0.086	0.012	0.096
NC07	7.2	0.10	0.28	0.14	0.38	0.11	0.11	0.15	0.20
NC03	8.6	0.13	0.42	0.19	0.59	0.28	0.19	0.33	0.31
NC06	8.8	0.064	0.051	0.13	0.073	0.23	0.078	0.32	0.10
NC05	9.4	0.099	0.44	0.22	0.67	0.27	0.20	0.39	0.36
NP07	9.7	0.003	0.093	0.004	0.12	0.003	0.036	0.004	0.063
NC01	11.0	0.14	0.35	0.22	0.46	0.35	0.19	0.44	0.26
NP06	12.0	0.004	0.048	0.007	0.051	0.005	0.030	0.008	0.037
NP05	13.0	0.006	0.17	0.012	0.23	0.010	0.10	0.015	0.14
NC09	13.0	0.12	0.15	0.18	0.19	0.23	0.11	0.30	0.14
NP09	15.0	0.004	0.18	0.006	0.22	0.007	0.097	0.009	0.13
NC08	15.0	0.080	0.065	0.16	0.081	0.29	0.11	0.44	0.14
NP03	16.0	0.002	0.036	0.004	0.38	0.004	0.037	0.005	0.040
PC3	16.0	0.005	0.071	0.008	0.22	0.009	0.46	0.012	0.86
NP08	17.0	0.005	0.086	0.011	0.10	0.012	0.092	0.017	0.10
NC02	17.0	0.18	0.19	0.31	0.22	0.51	0.19	0.65	0.22
CP3	17.0	0.14	0.23	0.28	0.30	0.57	0.22	0.84	0.30
CP4	17.0	0.14	0.27	0.28	0.45	0.56	0.23	0.84	0.33
NC10	18.0	0.038	0.067	0.073	0.082	0.11	0.049	0.16	0.063
NP01	18.0	0.004	0.28	0.005	0.42	0.003	0.092	0.004	0.21
NP10	18.0	0.003	0.042	0.005	0.049	0.004	0.027	0.006	0.032
BC3	19.0	0.071	0.13	0.089	0.22	0.094	0.050	0.12	0.11
CP2	20.0	0.12	0.086	0.30	0.12	0.59	0.20	0.90	0.29
NP02	29.0	0.003	0.15	0.004	0.20	0.003	0.054	0.004	0.10
BF5	33.0	0.081	0.20	0.93	0.35	0.078	0.064	0.091	0.17
BF8	33.0	0.081	0.20	0.93	0.35	0.078	0.064	0.091	0.17
PC2	38.0	0.003	0.058	0.003	0.14	0.003	0.22	0.004	0.57
PC1	71.0	0.020	0.29	0.029	0.43	0.028	0.77	0.039	1.11
BC1	230.0	0.094	0.17	0.12	0.22	0.10	0.096	0.14	0.16
CP1	380.0	0.60	1.7	0.85	1.8	1.3	1.0	1.6	1.3
BF7	1,900	0.12	0.79	0.23	1.2	0.37	0.29	0.51	0.61
BF6	23,000	0.42	1.3	0.59	1.6	0.50	0.58	0.61	0.94

a. All figures of merit for case BP1, BP2, BP3, BF1, BF2, BF3, and BF4 were less than 0.01

Bibliography

1. Bridgman, C.J. Class notes distributed for NENG605 and NENG631, Nuclear Explosives and Prompt Effects of Nuclear Weapons. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, 1987.

2. Burden, Richard L. and J. Douglas Faires. *Numerical Analysis* (Third Edition). Boston: Prindle, Weber and Schmidt, 1985.

3. Fletcher, R. and M.J.D. Powell. "A Rapidly Convergent Descent Method for Minimization," *Computer Journal* 6: 163-168 (July 1963).

4. Gorbachenko, G.M. et al. "Use of Fluorescence in the Spectrometry of Pulsed X Radiation," *Instruments and Experimental Technology* , 19: 244-246 (Jan-Feb 1976).

5. Hewlett-Packard Company. *Curve Fitting Pac Owner's Manual for the HP-71.* Singapore: Hewlett-Packard Company, 1984.

6. Mathews, K.A., Air Force Institute of Technology. Private communication. (1987).

7. National Bureau of Standards. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables,* edited by Milton Abramowitz and Irene A. Stegun. Applied Mathematics Series; No., 55. Washington: Government Printing Office, 1964.

VITA

Captain Russell B. Daniel was born on 26 July 1960 in Tulia, Texas. He graduated from Tulia High School in 1978 and entered the United States Naval Academy. Captain Daniel received a Bachelor of Science in Marine Engineering and was commissioned a Second Lieutenant in the United States Marine Corps in May of 1982. Following completion of The Basic School and The Infantry Officer's Course at Quantico, Virginia, Captain Daniel was assigned to 2nd Battalion, 3rd Marines (Rein). Captain Daniel served with the 2nd Battalion, 3rd Marines (Rein) from September 1983 to April 1985. During this time, he served as assistant 81mm mortar platoon commander, rifle platoon commander, and as executive officer for the weapons company. Captain Daniel was then transferred to the 3rd Marines (Rein) operations office. He served on the regimental staff until being selected for the Marine Corps special education program and entering the Air Force Institute of Technology in August of 1986.

Permanent Address: Box 94
Tulia,Texas 79088

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GNE/ENG/88M-3	5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION School of Engineering	6b OFFICE SYMBOL (If applicable) AFIT/ENP	7a NAME OF MONITORING ORGANIZATION
6c ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB OH 45433-6583		7b ADDRESS (City, State, and ZIP Code)
8a NAME OF FUNDING/SPONSORING ORGANIZATION Defense Nuclear Agency	8b OFFICE SYMBOL (If applicable) TTTT	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c ADDRESS (City, State, and ZIP Code)
Kirtland AFB, New Mexico 87115-5000

10 SOURCE OF FUNDING NUMBERS N/A

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)

12 PERSONAL AUTHOR(S)
Russell E. Daniel

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
MS Thesis	FROM ________ TO ______	1988 March	149

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP
18	04	

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
X-ray Diagnostics
Fredholm Integral Equations

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: AN APPROXIMATION TECHNIQUE FOR SOLVING A SYSTEM
OF FREDHOLM INTEGRAL EQUATIONS FOR ASYMMETRIC
DETECTOR RESPONSE FUNCTIONS

Thesis Chairman: Kirk A. Mathews, LCDR, USN
Associate Professor of Nuclear Engineering (Effects)

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (Include Area Code)	22c OFFICE SYMBOL
Kirk A. Mathews, LCDR, USN	(513)255-3030	AFIT/ENG

DD Form 1473, JUN 86 Previous editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Abstract

The purpose of this study was to develop the methodology for and to implement a computer program to approximate a solution to a system of Fredholm integral equations. The system of equations used in this study is representative of the equations formed during the detection of pulsed radiation using a series of detectors with asymmetric response functions. Though general in nature and applicable to all systems of fredholm integral equations, the equations studied are of importance to the Defense nuclear Agency with regard to the measurement of radiation spectra during underground nuclear effects simulation testing.

The deconvolution technique consisted of representing the unfolded spectrum as a weighted sum of basis functions. This unfolded spectrum, the actual spectrum, and a predicted spectrum were then used to form a X^2 test statistic. By adjusting the parameters in the basis functions and their weights, X^2 was minimized and the unfolded spectrum was corrected to approximate the actual spectrum.

The methodology for this deconvolution technique was then converted into a general computer program. The validation cases conducted on the two types of spectra confirmed the reliability of the methodology and the computer program. Additionally, an initial study with simulated measurement error added to the measured-to-predicted ratios showed that the actual spectrum could not be returned exactly. The second study approximated the actual spectrum with an unfolded spectrum using a second set of basis functions. An acceptable approximation was conducted; however, certain artifacts were discovered in the unfolded spectrum. The validation cases and preliminary test cases conducted prove that the computer program based on the methodology presented in this study is a viable means of approximating an actual radiation spectrum.

END
DATE
FILMED

4-88
DTIC